C= újság

Az Országos Commodore Egyesület lapja

1989/6



JÁTÉK
TIPPEK —
TRÜKKÖK
SZÁMOLÓ
COMMODORE
KÖNYVEK
VIGYÁZAT,
VÍRUS!

AGFA X 41 másológép
raktárról azonnal megvehető
Ára: 500 800,– Ft + 25% ÁFA

Nagyít, kicsinyít 70, 80 és 130%-os léptékben
Legnagyobb másolható méret A3
Automata lapadagoló RDF
100 000 másolatra anyag az árban
Tűzőgép
12 havi vagy 100 000 másolatig garancia
40 másolat percenként
Garancián túli szerviz biztosítva
Alkatrészek KO-raktárról forintért

1033 Budapest,
Szentendrei út 89.
Tel: 805-746, 805-618

1031 Budapest,
Kaszás dűlő 1-3.
Telefon: 805-587,
805-565, 805-278.
Telex: 22 7337.

Agro-Industria
INNOVÁCIÓS VÁLLALAT

ORSZÁGOS PC SZERVIZHÁLÓZAT!

IBM-kompatíbilis PC-k,
házi számítógépek szakszerű,
gyors javítása!
Hálózatkialakítás, bővítések, illesztések,
gépösszeszerelés.
Gyártók és forgalmazók garanciális kötele-
zettségeit átvállaljuk.

Bárhol az ország területén 48 órán belül megjelenünk
a hiba elhárítására!



Kirendeltségek:

Győr	Babits M. utca 8/B.	20-947.
Székesfehérvár	Móri utca 58.	16-814.
Zalaegerszeg	Bíró M. utca 14/A.	13-789.
Szeged	Retek utca 23—25.	25-448.
Debrecen	Katona J. utca 1/A.	34-534.
Miskolc	Huba utca 23.	89-308

Berendezéseink megbízható működése érdekében

LEGYEN AZ ÜGYFELÜNK!

AGROINDUSTRIA

MIT, HOGYAN, HOL, MIKOR?

EGYESÜLETI ÜGYEK: Egyesületünk tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az Egyesület irodájában (1133 Budapest, Kárpát u. 7/a. I. em. 11., tel: 497-559), vagy átutalással az MNB 217-98292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén számlát küldünk.

Minden tagunk rendelkezésére áll PÖTYÖGŐSZOLGÁLATUNK, a szervizkedvezmény és az apróhirdetési lehetőség.

A Deákpáholy tagjai minden hónapban megkapják a C-újságot, a tagsági díj fél évre 366 forint.

A Pluszpáholy tagjai minden hónapban megkapják a C-újságot, és minden hónapban 2×60 forint vásárlási utalványt is kapnak. A tagsági díj fél évre 911 forint.

A Szuperpáholy tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 1800 forint vásárlási utalványt. A féléves tagsági díj 10 000 forint.

ÜGYFÉLFOGADÁS: minden kedden és csütörtökön 14—18 óra között várjuk tagjainkat és az érdeklődőket.

PÖTYÖGŐSZOLGÁLAT: Az újságban megjelenő programokat másolja a megrendelők részére. Megrendelhető személyesen az Egyesület irodájában, vagy postai utánvéttel.

A pötyögőszolgálat minden páros héten csütörtökön az egyesületi irodában működik, 16—19 óra között.

APRÓHIRDETÉS: Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 80 forint. A hirdetés módja: az újságban megjelenő nyomtatvány kitöltésével.

A C-újság régebbi számai megvásárolhatók az Egyesület irodájában, vagy megrendelhetők utánvéttel.

Kedvezményes ár! Tagoknak olcsóbb!

Az újságban eddig megjelent programok gépenként összegyűjtve megrendelhetők. VC 20, C16, PLUS/4, C128, C64. További felvilágosítást is adunk a 497-559-es telefonszámon, vagy levélben!

Vidéki Pluszpáholy-tagjaink három havi tikett összegyűjtésekor igénybe vehetik a NOVOTRADE 2C áruház ingyenes csomagküldő szolgálatát.

Vidéken további információk kaphatók:
Jászberényi Városi Könyvtár
Győri Bartók Béla Művelődési Ház
Zalaegerszegi Ságvári Endre Gimnázium
Pécsi Apáczai Csere János Gimnázium

Budapesten havonta klubdélelőtt a Petőfi Csarnokban.

FIGYELEM!
Kérjük a második félévi tagdíjakat befizetni!

A kedvezmények a következő vidéki könyvesboltok 2C sarkaiban válthatók be.

PÉCS: Zrínyi Miklós Könyvesbolt. 7621 Jókai u. 25. Tel.: 72-12835
DEBRECEN: Szak- és ismeretterjesztő Könyváruház. 4024 Hunyadi u. 8. Tel.: 52-23237
SZOMBATHELY: Savaria Könyvesbolt. 9700 Mártírok tere 1. Tel.: 94-12341
VESZPRÉM: Kölcsey Ferenc Könyvesbolt. 8200 Cserhát út 7.
BÉKÉSCSABA: Radnóti M. Könyvesbolt. 5600 Tanácsköztársaság út 2. Tel.: 25-207
GYŐR: Pattantyús Á. Géza Szakkönyvesbolt. 9021 Molnár Ferenc u. 9.
SZEGED: Tömörkény Könyvesbolt. 6720 Lenin krt. 48. Tel.: 62-21453
SZOLNOK: Szigligeti Könyvesbolt. 5000 Ságvári krt. 35. Tel.: 56-11133
MISKOLC: Chip-kuckó. 3530 Tanácsház tér 14.

Egyesületünk és az újság szerkesztősége várja az Önök programjait, hardver leírásait, cikkeit. Szerkesszük együtt az újságot!

FIGYELEM! FIGYELEM!

Értesítjük Egyesületünk tagjait és a Commodore Újság minden olvasóját, hogy következő számunk, a július—augusztusi, összevont számként jelenik meg, ára 96 Ft.
Tagságunk, valamint az Olvasók augusztus első hetében juthatnak hozzá.

Szerkesztőség

Az Országos Commodore Egyesület módszertani kiadványa.

Egyesületi iroda és szerkesztőség: 1133 Budapest, Kárpát u. 7/a I. em. 11. Tel.: 497-559
Felelős kiadó: Horváth Judit, az egyesület elnöke
Főszerkesztő: Rados Péter
Felelős szerkesztő: Dr. Horváth András
Művészeti szerkesztő: Szulyovszky József
Egyesületi szervező: Winter Júlia
Lapmenedzser: Kovács Gábor
Levélcím: Commodore Újság, 1388 Bp. 62. Pf. 86.
Index: ISSN 0237-756 X
Terjeszti a Magyar Posta.
Megvásárolható a hírlapárusoknál.
MSZH Nyomda

Tisztelt Commodore Újság!

Most egy újabb, C16-ra és Plus/4-re íródott programot ajánlok, amely hosszú BASIC programok írása közben nyújthat segítséget a felhasználónak.

A Színbeállító lényege az, hogy BASIC program írása előtt betöltve, lényegesen egyszerűbbé válik a színek megválasztása. Nem kell sok-sok COLOR paranccsal próbálgatni a leglátványosabb, legmegfelelőbb színösszeállítást. Az F1-keret, F2-háttér és F3-karakter billentyűk segítségével változtathatjuk a színeket, és a legutoljára lenyomott funkciónak megfelelő helyen a HELP billentyű segítségével beállíthatjuk a megfelelő árnyalatot. Eközben pedig a képernyő felső részén kialakított ablakban megjelennek a' aktuális színeknek megfelelő COLOR utasításhoz írandó színkódok.

A programot eredetileg gépi kódban írtam, de átírtam BASIC-be DATA sorokba rendezve. Futtatáskor a verem nem használt területére és a beépített ROM-ok indítóprogramjai területére másolódik és a NEW hatására törlődik a BASIC betöltő. Mellékelten elküldöm a program Assembly listáját is.

Üdvözlettel:

Igali István

```
1 REM **********************************
2 REM *      C= UJSAG          SORSZAM:159   *
3 REM *      SZINBEALLITO C16 - PLUS/4     *
4 REM *      PROGRAM: IGALI ISTVAN          *
5 REM **********************************
10 KEY1,"▒"+CHR$(27)+"T▒▒▒▒▒▒"+CHR$(27)+"BOS 11525"+CHR$(13)+"▒▒"
15 KEY2,"▒"+CHR$(27)+"T▒▒▒▒▒▒"+CHR$(27)+"BOS 11554"+CHR$(13)+"▒▒"
20 KEY3,"▒"+CHR$(27)+"T▒▒▒▒▒▒"+CHR$(27)+"BOS 11583"+CHR$(13)+"▒▒"
25 KEY8,"▒"+CHR$(27)+"T▒▒▒▒▒▒"+CHR$(27)+"BOS 11612"+CHR$(13)+"▒▒"
30 FOR I=275 TO 370:READ D$
35 A=DEC(D$):X=X+A
40 POKE I,A:NEXT I
45 FOR I=1525 TO 1724:READ D$
50 A=DEC(D$):X=X+A
55 POKE I,A:NEXT I
60 IF X<>29749 THEN 75
65 POKE 208,14:POKE 211,6
70 POKE 209,1:POKE 212,7:NEW
75 PRINT"HIBA AZ ADATOKBAN!"
100 DATA A0,00,A2,00,BD,D0,00,C9
110 DATA 09,10,09,18,69,31,99,24
120 DATA 0C,4C,34,01,48,A9,31,99
130 DATA 23,0C,68,18,69,27,99,24
140 DATA 0C,BD,D3,00,18,69,30,99
150 DATA 26,0C,98,18,69,28,A8,E8
160 DATA E0,03,D0,D0,A2,34,8E,20
170 DATA 0C,A2,30,8E,48,0C,A2,31
180 DATA 8E,70,0C,A2,00,AD,3B,05
190 DATA 9D,00,08,E8,E0,78,30,05
200 DATA A9,44,9D,00,0C,E0,9F,D0
210 DATA EC,AD,3B,05,8D,9F,08,60
220 DATA A9,04,8D,B4,07,AE,D0,00
230 DATA E0,0F,D0,02,A2,FF,E8,8E
240 DATA D0,00,8E,19,FF,A0,00,8C
250 DATA D3,00,4C,13,01,A9,00,8D
260 DATA B4,07,AE,D1,00,E0,0F,D0
270 DATA 02,A2,FF,E8,8E,D1,00,8E
280 DATA 15,FF,A0,00,8C,D4,00,4C
290 DATA 13,01,A9,01,8D,B4,07,AE
300 DATA D2,00,E0,0F,D0,02,A2,FF
310 DATA E8,8E,D2,00,8E,3B,05,A0
320 DATA 00,8C,D5,00,4C,13,01,AE
330 DATA B4,07,E0,04,D0,20,AD,19
340 DATA FF,18,69,10,C9,80,10,03
350 DATA AD,00,00,8D,19,FF,AC,D3
360 DATA 00,C8,C0,08,D0,02,A0,00
370 DATA 8C,D3,00,4C,13,01,E0,00
380 DATA D0,20,AD,15,FF,18,69,10
390 DATA C9,80,10,03,AD,D1,00,8D
400 DATA 15,FF,AC,D4,00,C8,C0,08
410 DATA D0,02,A0,00,8C,D4,00,4C
420 DATA 13,01,AD,3B,05,18,69,10
430 DATA C9,80,30,03,AD,D2,00,8D
440 DATA 3B,05,AC,D5,00,C8,C0,08
450 DATA D0,02,A0,00,8C,D5,00,4C
460 DATA 13,01,00,00,00,00,00,00

READY.
```


HÍREK

BASIC-bővítés a PLUS/4-eshez

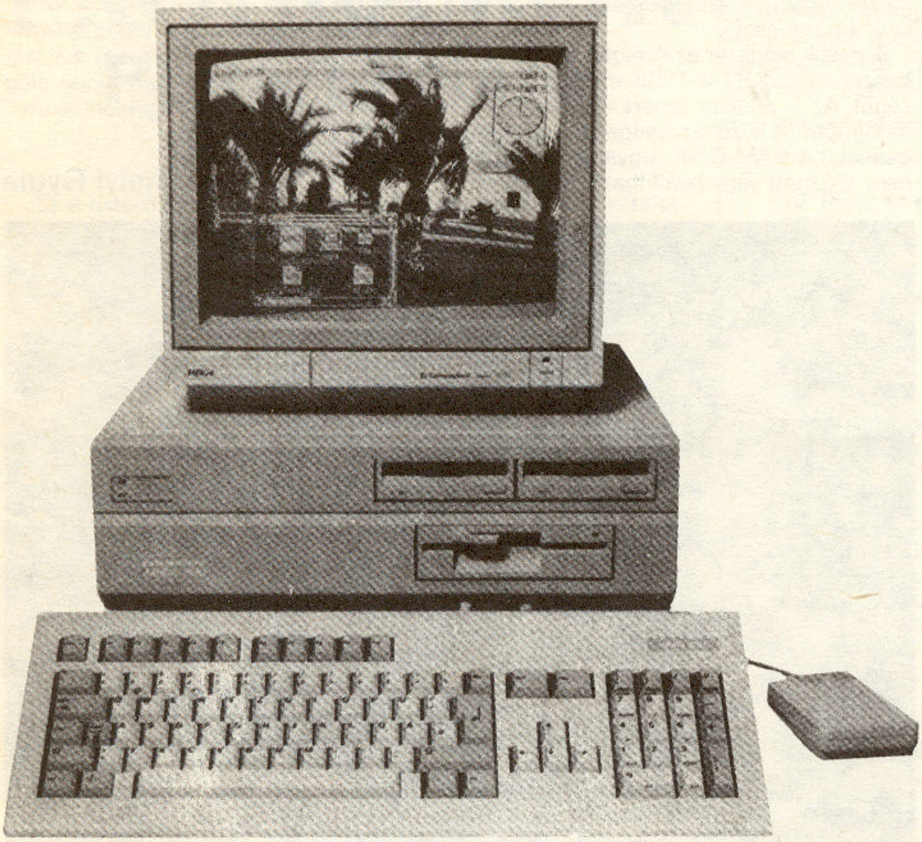
A Sven Giero Datentechnik egy új BASIC bővítést kínál a legalább 32 kbyte RAM-mal felbővített C16-hoz vagy a PLUS/4-eshez. A lemezen kapható program neve Professional BASIC, s az több, mint 80 új parancsot tartalmaz.

A bővítés számos segítséget tartalmaz, amelyekkel a BASIC programok editálása könnyebb lehet. A Professional BASIC német kézikönyvvel együtt kb. 45 márkáért kapható.

Info.: Sven Giero Datentechnik. Lessingstrasse 14, D—3167 Burgdorf.

Az AMIGA operációs rendszere

I. rész



Ezzel az írással elsősorban azoknak az Amiga-tulajdonosoknak szeretnék segíteni, akiknek nehézséget okoz az Amiga operációs rendszerének megértése, használata. Mivel a tapasztalat azt mutatja, hogy nem csak azoknak van ilyen jellegű problémájuk az Amigával, akiknek előzőleg egyszerűbb gépük (pl. Commodore 64) volt, ezért bárki másnak is, akit érdekel a téma, hasznos lehet a cikk.

Felhasználói szempontból az operációs rendszer két fő részre osztható. Ennek a két résznek a neve Workbench és CLI. A két részrendszer nem független egymástól, sőt, szorosan kapcsolódnak egymáshoz, egymás mellett párhuzamosan is használhatók.

A Workbench egy egér által vezérelt, ikonokat, menüket és ablakokat használó rendszer. Hasonlít a Geos, a GEM vagy a Windows megfelelő környezetére. A Workbench különböző funkcióit az egér ill. a billentyűzet segítségével lehet elérni. Az egér által vezérelt — általában nyél alakú — jelet, szükség esetén a billentyűzetről is lehet mozgatni a jobb oldali Amiga + a kurzor mozgató billentyűkkel. A bal oldali egér-gombnak a bal oldali ALT + bal oldali Amiga — (egyes gépeknél Amiga helyett Commodore), a jobb oldali egér-gombnak a jobb oldali ALT + jobb oldali Amiga-billentyű felel meg. További hasznos billentyűkombinációk is vannak, pl.: az operációs rendszer kérdésére a 'Retry' választ a bal Amiga + v, a 'Cancel'-t a bal Amiga + b billentyűpárosokkal is megadhatjuk. Ha egy (vagy több) program saját képernyőt generál magának (tehát nem a Workbench képernyőt használja), akkor a program és a Workbench képernyői között a bal Amiga + n, bal Amiga + m gombokkal is átválthatunk.

A Workbench legfontosabb szolgáltatásai a különböző file-ok kezelésével kapcsolatosak; betöltés, kimentés, másolás, törlés, átnevezés, lemezformázás, az innen elérhető file-ok paramétereinek megjelenítése/módosítása, lemeztartalomjegyzék megjelenítése.

Nagyon fontos: Workbench-ből csak azok a file-ok érhetők el, amelyeknek vagy egy ugyanolyan nevű 'info' kiterjesztésű párja!!! Ugyanis itt nem lehet a hagyományos módon szöveges formában a katalógust megjeleníteni, csak ikonokkal. Mivel az ikonoknak sem az alakja, sem a mérete nincs rögzítve, ezért ennek adatait külön kell tárolni — többek között ezt tartalmazza az 'info' kiterjesztésű pár. Tehát ha pl. létezik egy 'API' nevű file a lemezen, akkor csak abban az esetben jelenik meg ez a Workbench-ben, ha van — ugyanabban a katalógusban — egy 'API.info' nevű file is, amely tartalmazza az 'API' ikonjának adatait (és még néhány más adatot. Ezekből következően egy lemezen akár több száz file is lehet, adott esetben ezekből csak néhányat vagy esetleg egyet sem láthatunk.

A legtöbb lemezen található egy 'Trashcan' nevű szemeteskosár alakú ikon. Ez első látásra törlésre való (ami nem kell, bedobjuk a szemétkosárba), valójában azonban egy közönséges alkatalógus ikonja. Tehát eltérően pl. a Geos-tól vagy a GEM-től, a szemétkosárba bedobott file-ok nem törlődnek, hanem átkerülnek ebbe a 'Trashcan' nevű alkatalógusba, s így nem vesznek el, bármikor visszamásolhatók eredeti helyükre. Ha azonban úgy döntünk, hogy a 'Trashcan' tartalmára biztos nem lesz szükségünk, a megfelelő szemetes kijelölése után az 'Empty Trash' menüpont segítségével „egyetlen mozdulattal" valóban törölhetjük a szemetesben lévő file-okat. Lehetőség van a 'Trashcan' — mint átmeneti tárolóhely — használata nélkül is, közvetlenül törölni a 'Discard' menüponttal.

Kevesen tudják, hogy hogyan lehet pl. több ikont kijelölni, és rajtuk egyszerre elvégezni bizonyos műveleteket. A probléma a következő: ha kijelölök egy ikont és egy másikat is ki szeretnék jelölni, abban a pillanatban, ahogy a másodikat kijelölöm, az első nyomban visszavált eredeti állapotába, tehát egyidőben egyszerre csak egy ikon lehet kijelölve. Nem így van ez, ha használom a SHIFT gombot, ugyanis ha a SHIFT billentyű lenyomva tartása mellett jelölöm ki az ikonokat, akkor a korábbiak nem váltanak vissza. Ez hasznos lehet többszörös másolásnál, törlésnél stb., vagy ha azt akarjuk, hogy egy programhoz a hozzá tartozó adatfile is rögtön betöltődjön. Ez utóbbival hasonló hatás érhető el, ha használjuk az 'Info' menüpont után nyíló ablakban a 'DEFAULT TOOL' részt. Az itt nevével megadott futtatható programot a Workbench a betölteni kívánt adatfile létrehozó-programjának tekinti. (Az 'Info' ablakban megadott paramétereket az Amiga a megfelelő 'info' kiterjesztésű file-ba menti ki.) Ezért van az, hogy amikor Workbench-ből betöltünk egy BASIC programot — ami természetesen nem futtatható közvetlenül —, a gép nem áll le hibaüzenettel, hanem betölti az Amiga BASIC-et — hiszen a megfelelő 'info' kiterjesztésű file-ban szerepel a 'DEFAULT TOOL' rovatban megadott programnév (ebben az esetben az AmigaBASIC) —,

majd az AmigaBASIC-nek átadja a BASIC program nevét, ami a saját szerkesztőjébe betölti azt. Ide tartozik még a 'TOOL TYPES' rovat is. Itt paramétereket lehet megadni, amiket a Workbench az adott program betöltése után ad át. Mivel sem a paraméterek, sem azok száma, sem típusa nem rögzített, ezért ezt a lehetőséget csak olyan programok használják ki, amelyek írásánál tudatosan törekedtek erre. Ilyen pl. a Notepad vagy a Professional Page.

Szintén kevesen ismerik a 'Clean Up' és a 'Shapshot' menüpontok használatát. Ezekhez tudni kell, hogy az ikonok helye az őket tartalmazó ablakban nem rögzített, tetszőleges lehet, és ez akár kézzel, akár a gép által beállítható. Ha nem akarunk bajlódni a kézzel — ill. egérrel — való helyezgetéssel, és megfelel a gép által készített sorrend, akkor a megfelelő lemez vagy alkatalógus kiválasztása után a 'Clean Up' menüponttal rendezhetjük az egy ablakban lévő ikonokat. Vigyázat, az ikonok új pozíciói csak a memóriában lesznek átírva, a lemezen — az 'info' kiterjesztésű file-okban — nem változik semmi! Ahhoz, hogy ez a lemezen is átíródjon, a 'Shapshot' menüpontot kell használni.

Nagyon röviden ennyit a többfeladatos üzemmódot támogató Wokbench-ről.

A másik fontos részrendszer a CLI. A CLI a Command Line Interpreter vagy Command Line Interface angol szavak kezdőbetűiből képzett betűszó. Fordítása tehát parancs-sor értelmező. Ez a része az operációs rendszernek a hagyományos operációs rendszerek környezetére (pl. MS—DOS) hasonlít.

Formailag a CLI a Workbench képernyőn, egy közönséges ablakban fut. Workbench-ből is és a CLI ablakból is — a több feladatos üzemmód eredményeképp — tetszőleges számú új CLI ablakot lehet nyitni. A CLI-ben nincs semmilyen olyan jellegű megkötés, mint a Workbench-ben, itt a lemezen lévő összes file elérhető. A CLI-ben az egérnek sok szerepe nincs, itt az AmigaDOS parancsok segítségével lehet kommunikálni a géppel. Az 1.2-es verziójú AmigaDOS-ban minden parancs külső parancs, ami a kezdő felhasználónak roppant nehézségeket szokott okozni — főleg, ha csak egy lemezmeghajtójuk van —, s ez egyből el is szokta venni a kedvüket a CLI-től. Azért nem olyan forrón eszik a kását! Van néhány olyan megoldás, amivel kényelmesebbé tehető a CLI használata. Az első megoldás abban az esetben hasznos, ha csak néhány AmigaDOS műveletet szeretnénk elvégezni. Minden AmigaDOS parancsnak, ha paraméteréül egy kérdőjelet adunk meg, akkor nem fut le rögtön, hanem — egyfajta „Inelp" funkcióként — kiírja a neki megadható összes paramétert, majd addig vár, amíg meg nem adunk valamit. Nos, egy lemezmeghajtós rendszerben ezalatt a várakozási idő alatt ki lehet cserélni a Workbench lemezt arra a lemezre, amelyre érvényesíteni akarjuk a kívánt parancsot.

A másik módszer az Amigán nagyon jól hasznosítható RAM DISK-et hívja segítségül. Az — általam ismert — összes számítógéptípus közül az Amigán a legintelligensebb a RAM DISK, ugyanis a mérete nem rögzített vagy beállítható, hanem dinamikusan változik annak függvényében, hogy belemásolok, vagy törlök belőle. (Egyetlen korlát a szabad memória nagysága.) Az első módszer hátránya, hogy nagyobb munkáknál borzasztóan kényelmetlen az állandó lemezcsere. A második módszer ezt küszöböli ki. A munka kezdetén másoljuk be a RAM DISK-be azokat az AmigaDOS parancsokat, amelyekkel előreláthatólag dolgozni fogunk, majd adjunk ki egy 'PATH RAM:' parancsot. Ezentúl a RAM DISK-ben lévő parancsokért már nem fog a gép lemezhez nyúlni.

Tekintve, hogy a CLI-nek (és a Workbench-nek is) annyi és olyan összetett szolgáltatásai vannak, hogy a lap egy évi számainak teljes terjedelme sem lenne elég az ismertetésükre, ezért egyelőre ennyit a CLI-ről.

Újhelyi Gyula



HÍREK

A C128 és a CP/M

A Markt Technik könyvkiadó részlege most egy újabb „C128 Extra" terméket dobott piacra 49 márkáért. Az elsősorban a grafikával foglalkozik. Egy jelkészlet editor, egy nyomtatórutin és egy speciálisan a grafikus képek tömörítéséhez használható program található a csomagban.

A második lemez CP/M formátumú. A CP/M-mel eleddig semmilyen lehetőség nem volt a 80 karakteres képernyő nagyfelbontású grafikájának kihasználására. A C128 Extra Grafik 128 programja segítségével Turbo Pascalban problémamentesen készíthetünk szép képeket, rajzokat.

Info.: Markt et Technik Buchverlag, Hans-Pinsel-Str. 2, D—8013 Haar bei München.

Info.: Markt et Technik Verlag AG, Hans-Pinsel-Strasse 2, D—8013 Haar bei München.

Számítógépes szaküzletek, áruházak computer osztályai.

A megadott árak minden esetben a gyártóktól/forgalmazóktól származnak. Azokban a törvény által előírt Mehrwertsteuer benne foglaltatik. A piaci árak ettől eltérhetnek.



PROGRAMOK

PLUS/1 listázó PLUS/4-re

Ez a listázó program, mint sok más társa speciális feladatot old meg. Legfőbb célja a lemezen tárolt programokról közlésre, bemutatásra alkalmas, azaz jól áttekinthető listát készíteni. Jellemzői:

1. Alkalmas egy lemezoldalon levő programcsomag tagjainak folyamatos listázására.
2. Minden új program vagy minden új lap kezdetén a program nevét is kinyomtatjuk.
3. Egy lapra csak 54 sort nyomtatunk a címsort is beleszámítva.
4. A programsorok számozását a lap szélétől beljebb kezdve 9 karakteres mezőben jobbra igazítjuk.
5. A listában a BASIC alapszavak — utasítások, függvények, operátorok — kiemelésére NAGYBETŰT használunk. A változókat, szövegkonstansokat kisbetűkkel írjuk.
6. Az idézőjelben szereplő vezérlő karaktereket, melyeket a normál listában inverz karakterek jelölnek, hárombetűs emlékeztetővel helyettesítjük.
7. A program szerkezetének kiemelésére mind a DO, mind a FOR kezdetű ciklusok magját 2—2 hellyel beljebb igazítva írjuk.

Természetesen az így készített lista csak akkor mutat igazán, ha a program eleve is jól szerkesztett, nem tartalmaz zsúfolt sorokat. Az ilyen tömörített kódok különben sem alkalmasak arra, hogy egy-egy jól működő programot, vagy akár csak egy ügyes ötletet megvalósító programfogást közkinccsé tegyünk.

A program működése.

Első lépésként (200—260) egy rövid rutint töltünk be a kazetta puffer szabad területére:

```
.0332   A2 00   LDX   #$00
.0334   E6 D8   INC   $D8
.0336   D0 02   BNE   $033A
.0338   E6 D9   INC   $D9
.033A   A1 D8   LDA   ($D8, X)
.033C   60      RTS
```

Ennek a gépi kódú betétnek az a feladata, hogy a BASIC ROM területéről elhozzon egy karaktert. A rutin 0-ás lapú indexelt indirekt címzést használ. A ROM kijelölt címét a $D8—$D9 regiszterek tartalmazzák. A rutin minden SYS híváskor ezt a címet eggyel növeli, majd a ROM-ból elhozott karaktert az akkumulátorból a rendszer a $7F2 címre kimenti. Innen a BASIC program már a PEEK(.) függvénnyel át tudja venni az adatot. A rutinnal a ROM-ban a $818E címtől kezdődő táblázathoz tudunk hozzáférni, melyre a PEEK(.) függvény közvetlenül nem alkalmazható, ugyanis a PLUS/4-ben ez mindig a RAM megadott címének tartalmát adja át a hívó BASIC programnak. A szóbanforgó táblázat tartalmazza a BASIC alapszavakat a tokenek sorrendjében. Ezek a V3.5 verzióban 128-tól 253-ig terjednek.

A program második lépésében (300—330) a kétoszlopos K$(255, 1) táblázatot feltöltjük a megfelelő karakterekkel. Ezután (340—450) a ROM táblázatából átmásoljuk az alapszavakat a K$(...) táblázat 0 indexű oszlopába. A vezérlő kódok emlékeztetőit, melyeket a 460—560 sorokban adtunk meg, a táblázat 1 indexű oszlopába töltjük (570—600). A kódtáblázat feltöltése eléggé hosszadalmas, ezért e művelet idejére a képernyőkezelést felfüggesztjük (120 és 610 sorok). A program belövése idejére célszerű a következő két sort beírni:

435 PRINT K$(T, 0)' ';
595 PRINT K$(T, 1)' ';

A programcsomag elemszámának, a programok nevének majd a kezdő lapszámnak a beírása (1000—1070) után a listázás kezdődik (1120—1420). A program fontosabb változóinak szerepe a REM sorokban olvasható. Külön kell szólni a p$—p változókról. Ezek a listázott program lemezről beolvasott karakterét és ennek ASCII kódját jelölik. A BASIC sorok és a program végének figyelésén kívül ezzel vezéreljük azt, hogy idézőjelen kívüli q=0 vagy idézőjelen belüli q=1 kódot K$(p, q) nyomtassunk.

```
plus/1 lista

 10 REM"******************************
 20 REM"*  C=Újság sorszám: 160       *
 30 REM"*  PLUS/1 listázó PLUS/4-re   *
 40 REM"*    Dr. Hack Frigyes         *
 50 REM"******************************
 60 REM"*  CBM BASIC V3.5             *
 70 REM"*  C=16   -  PLUS/4           *
 80 REM"*  VIC 1541 / 1551            *
 90 REM"*  MPS-801/803 - SP-180VC     *
100 REM"******************************
110 DIM k$(255,1)
120 POKE 65286,PEEK(65286) AND 239
130 :
200 REM"=== Gépi kód betöltése =======
210 POKE DEC("d8"),DEC("8d")
220 POKE DEC("d9"),DEC("81")
230 DATA a2,0,e6,d8,d0,2,e6,d9,a1,d8,60
240 FOR m=DEC("332") TO DEC("33c")
250   READ c$ : POKE m,DEC(c$)
260 NEXT
270 :
300 REM"=== Kódtáblázatok ============
310 FOR i=0 TO 255
320   k$(i,0)=CHR$(i):k$(i,1)=CHR$(i)
330 NEXT
340 REM"--- Tokenek:128-253 ----------
350 t=128
360 DO WHILE t<254
370   t$=CHR$(145)
380   DO
390     SYS DEC("332")
400     b=PEEK(DEC("7f2"))
410     t$=t$+CHR$(b AND 127)
420   LOOP WHILE b<128
430   k$(t,0)=t$+CHR$(17)
440   t=t+1
450 LOOP
460 REM"--- Vezérlő kódok ------------
470 DATA   5,[wht], 13,[rtn], 17,[cud]
480 DATA  18,[rvs], 19,[hom], 20,[del]
490 DATA  28,[red], 29,[cur], 30,[grn]
500 DATA  31,[blu],129,[org],130,[flo]
510 DATA 132,[flf],144,[blk],145,[cup]
520 DATA 146,[rvo],147,[clr],148,[ins]
530 DATA 149,[brn],150,[ygn],151,[lrd]
540 DATA 152,[bgn],153,[lbl],154,[dbl]
550 DATA 155,[lgn],156,[pur],157,[cul]
560 DATA 158,[yel],159,[cyn],   0,[-]
570 DO
580   READ t : IF t=0 THEN EXIT
590   READ k$(t,1)
600 LOOP
610 POKE 65286,PEEK(65286) OR 16
```

```
plus/1 lista

1000 REM"=== Adatbevitel ============
1010 INPUT "[clr][cud] Programok száma=";n
1020 DIM n$(n)
1030 FOR j=1 TO n
1040   PRINT USING "###. program=";j;
1050   INPUT n$(j)
1060 NEXT j
1070 INPUT "[cud] Kezdő lapszám=";l
1080 s=0 : REM"Sorszámláló
1090 OPEN15,8,15,"i"
1100 OPEN 4,4,7
1110 REM"< L-i-s-t-á-z-ó  c-i-k-l-u-s >
1120 FOR j=1 TO n
1130   m=10          : REM"Baloldali margó
1140   OPEN 3,8,3, n$(j)+",p,r"
1150   IF s>50 THEN GOSUB 3000
1160   GOSUB 2000
1170   GET#3,p$,p$,p$,p$ : p=ASC(p$)
1180   DO UNTIL p=0
1190     IF s>54 THEN GOSUB 3000
1200     IF s= 0 THEN GOSUB 2000
1210     GET#3,lb$,hb$ : REM"BASIC sorszám
1220     lb=ASC(lb$+CHR$(0))
1230     hb=ASC(hb$+CHR$(0))
1240     ln=256*hb+lb
1250     PRINT#4,USING "######### ";ln;
1260     q=0 :GET#3,p$: p=ASC(p$)
1270     IF p=236 OR p=130 THEN m=m-2
1280     m$=RIGHT$(STR$(100+m),2)
1290     PRINT#4,CHR$(16)m$;
1300     DO UNTIL p=0
1310       PRINT#4,k$(p,q);
1320       IF p=235 OR p=129 THEN m=m+2
1330       IF p=34 THEN q=1-q
1340       GET#3,p$ : p=ASC(p$)
1350     LOOP
1360     PRINT#4 : s=s+1
1370     GET#3,p$,p$   : p=ASC(p$)
1380   LOOP
1390   CLOSE 3 :PRINT#4 : s=s+1
1400 NEXT j
1410 CLOSE 4 : CLOSE 15
1420 END
2000 REM"---- Fejléc + lapszám ----
2010 PRINT#4,CHR$(14)"    "n$(j)CHR$(15);
2020 ll$=CHR$(16)+"45-"+STR$(l)+" -"
2030 IF s=0 THEN PRINT#4,ll$; :l=l+1
2040 PRINT#4 :PRINT#4 :s=s+2 :RETURN
3000 REM"---- Lapdobás --------
3010 DO WHILE s<72
3020   PRINT#4 :s=s+1
3030 LOOP :s=0 :RETURN
```

(Az idézőjel ASCII kódja 34.) A p aktuális értékéből állapítjuk meg az adott sorban szereplő DO=235 és FOR=129 illetve LOOP=236 és NEXT=130 alapszavakat. Ha ilyen szerepel, akkor az m margó méretét a megfelelő irányban 2-vel változtatjuk.

Fejlesztési lehetőségek

A közölt program szándékosan csak a legszükségesebb funkciókat valósítja meg, alkalmat adva az egyéni ízlés és szükséglet szerinti kiegészítésre. A fejlesztés során az első lépéshen a megfelelő adatvédelmet célszerű megvalósítani. További ajánlott kiegészítés a lapszámozás, címkiírás feltételessé tétele, vagy a sorok számának megadása. Több lemezre is kiterjeszthetjük a listázást. Ekkor a lemezcsere idejére várakoztatni kell a programot. Ilyen várakoztatásra esetleg laponként, vagy programonként is sor kerülhet. A vezérlő karakterek emlékeztető kódjainak táblázata is egyénileg változtatható. A legtöbb módosítást azoknak ajánlhatjuk, akik az MPS801-nél többre képes nyomtatóval dolgoznak. A program 370 és 430 soraiban állítjuk be a BASIC alapszavak nyomtatási típusát. A CHR$(145) kód a nagybetűs készletre váltja át a nyomtatót, a CHR$(17) pedig visszaállítja az SA=7 másodlagos címmel megnyitott kisbetűs írást. Ezeket a kódokat kicserélhetjük más írástípust vezérlőkre. A program listája SP—180VC nyomtatóval készült úgy, hogy a sorszámokat dőlt betűkkel, a BASIC alapszavakat pedig kövér betűkkel nyomtattuk.

Dr. Hack Frigyes

FAST VALIDATE C+4

Az itt közölt program a C- újság 1988/9. számában megjelent C64-es program átirata PLUS/4-es gépre. Beírás után mentsük ki a programot, majd futtassuk. A gépi kódú programot saját magára tölti. Ezt ismét mentsük ki. (A program beállítja a $2B—$2C és a $2D—$2E vektorokat, ezért DSAVE-vel kimenthető.) A működési leírást lásd a C64-es változatnál.

```
100 REM"*******************************
105 REM"*                             *
110 REM"* C= UJSÁG         SORSZÁM:161 *
115 REM"* FAST VALIDATE +4            *
120 REM"* PROGRAM CROWN S. CO.        *
125 REM"*                             *
130 REM"*******************************
135 REM
140 DATA 0,36,16,195,7,158,52,49
145 DATA 51,52,32,70,65,83,84,45
150 DATA 86,65,76,73,68,65,84,69
155 DATA 32,32,67,61,32,85,74,83
160 DATA 65,71,32,0,0,0,32,11
165 DATA 142,234,234,234,234,234,234,234
170 DATA 234,234,234,234,234,234,234,234
175 DATA 234,234,234,234,234,234,234,234
180 DATA 234,234,234,234,234,234,234,234
185 DATA 234,234,234,234,234,234,234,234
190 DATA 234,234,234,32,231,255,169,192
195 DATA 141,25,255,169,193,141,21,255
200 DATA 169,113,141,59,5,169,135,160
205 DATA 20,32,144,18,32,228,255,240
210 DATA 251,201,3,208,3,76,75,17
215 DATA 169,49,160,21,32,144,18,169
220 DATA 1,162,8,160,15,32,186,255
225 DATA 169,0,133,183,32,192,255,32
230 DATA 214,17,185,161,18,32,210,255
235 DATA 200,208,247,169,51,32,196,17
240 DATA 32,214,17,185,11,19,32,210
245 DATA 255,200,208,247,169,0,162,64
250 DATA 133,251,134,252,169,35,133,255
255 DATA 169,51,32,196,17,32,204,255
260 DATA 162,1,32,198,255,160,0,32
265 DATA 207,255,145,251,206,21,255,238
270 DATA 21,255,200,192,42,144,240,24
275 DATA 165,251,105,64,133,251,144,2
280 DATA 230,252,32,225,255,240,90,169
285 DATA 56,198,255,208,205,32,204,255
290 DATA 32,214,17,185,205,19,32,210
295 DATA 255,200,192,144,144,245,162,1
300 DATA 138,10,10,168,169,255,153,1
305 DATA 112,153,2,112,138,32,14,18
310 DATA 153,0,112,56,233,25,73,255
315 DATA 170,169,255,74,202,208,252,153
320 DATA 3,112,166,34,232,224,36,144
325 DATA 215,162,18,160,1,142,64,68
330 DATA 140,65,68,136,32,30,18,169
335 DATA 52,32,196,17,32,225,255,208
340 DATA 20,162,1,32,201,255,169,73
345 DATA 32,210,255,234,234,234,234,234
350 DATA 234,234,76,180,17,32,204,255
355 DATA 162,1,32,198,255,32,207,255
360 DATA 201,255,208,73,32,207,255,201
365 DATA 255,240,23,170,32,207,255,168
370 DATA 32,30,18,32,207,255,170,32
375 DATA 207,255,168,32,30,18,169,51
380 DATA 208,183,32,214,17,185,74,20
385 DATA 32,210,255,200,192,36,144,245
390 DATA 160,4,185,0,112,32,210,255
395 DATA 200,192,144,144,245,169,51,32
400 DATA 196,17,32,204,255,162,1,32
405 DATA 198,255,32,207,255,32,210,255
410 DATA 201,13,208,246,32,204,255,169
415 DATA 1,32,195,255,169,2,32,195
420 DATA 255,76,124,18,72,32,204,255
425 DATA 162,1,32,201,255,169,85,32
430 DATA 210,255,104,76,210,255,32,204
```

```
435 DATA 255,169,2,32,195,255,169,2
440 DATA 162,8,160,2,32,186,255,169
445 DATA 2,162,110,160,20,32,189,255
450 DATA 32,192,255,162,1,32,201,255
455 DATA 160,7,185,111,20,32,210,255
460 DATA 136,208,247,32,204,255,162,2
465 DATA 32,201,255,160,0,96,134,34
470 DATA 162,4,221,118,20,202,176,250
475 DATA 189,123,20,166,34,96,120,224
480 DATA 36,176,250,138,240,247,32,14
485 DATA 18,133,34,196,34,176,238,134
490 DATA 34,132,35,138,10,10,133,36
495 DATA 152,74,74,74,24,101,36,170
500 DATA 152,41,7,168,185,127,20,61
505 DATA 1,112,240,209,185,127,20,73
510 DATA 255,61,1,112,157,1,112,166
515 DATA 36,222,0,112,165,35,10,168
520 DATA 198,34,165,34,162,0,134,34
525 DATA 74,102,34,74,102,34,9,64
530 DATA 133,35,177,34,170,200,177,34
535 DATA 168,76,30,18,88,96,0,0
540 DATA 0,0,0,0,0,0,0,0
545 DATA 0,0,0,0,0,0,0,0
550 DATA 133,34,132,35,160,0,177,34
555 DATA 240,6,32,210,255,200,208,246
560 DATA 96,160,0,185,12,5,153,0
565 DATA 4,200,208,247,96,24,144,51
570 DATA 24,144,73,133,74,174,0,28
575 DATA 36,74,48,2,232,232,202,138
580 DATA 41,3,133,75,173,0,28,41
585 DATA 252,5,75,141,0,28,162,0
590 DATA 169,16,202,208,253,74,144,250
595 DATA 198,74,165,74,41,127,208,213
600 DATA 134,74,96,169,255,141,5,24
605 DATA 44,5,24,16,12,44,0,28
610 DATA 48,246,173,1,28,184,160,0
615 DATA 96,76,168,5,133,3,165,3
620 DATA 48,252,201,1,240,242,162,3
625 DATA 76,10,230,32,66,208,120,169
630 DATA 162,32,6,4,169,1,133,12
635 DATA 208,11,120,32,126,249,230,12
640 DATA 169,2,32,6,4,173,12,28
645 DATA 9,14,141,12,28,169,255,160
650 DATA 44,153,255,2,136,208,250,132
655 DATA 48,132,49,200,132,60,165,12
660 DATA 32,75,242,133,59,133,55,138
665 DATA 10,10,10,10,10,133,68,173
670 DATA 0,28,41,159,5,68,141,0
675 DATA 28,32,0,4,80,254,184,173
680 DATA 1,28,201,82,240,6,198,60
685 DATA 16,239,48,71,162,2,80,254
690 DATA 184,173,1,28,202,16,247,42
695 DATA 8,74,74,74,168,185,160,248
700 DATA 40,106,74,74,133,70,32,0
705 DATA 4,162,0,80,254,184,173,1
710 DATA 28,149,36,232,224,6,144,243
715 DATA 160,36,32,232,247,165,70,168
720 DATA 74,197,59,176,10,165,83,153
725 DATA 0,3,165,84,153,1,3,198
730 DATA 55,208,166,173,0,3,162,43
735 DATA 141,67,2,142,73,2,32,44
740 DATA 193,120,169,1,133,165,169,3
745 DATA 133,166,76,86,230,24,144,24
750 DATA 120,169,164,32,6,4,88,32
755 DATA 66,208,169,2,133,59,169,6
760 DATA 133,60,162,18,160,1,208,31
765 DATA 174,0,6,172,1,6,24,165
770 DATA 59,105,32,133,59,144,27,36
775 DATA 27,48,5,169,144,32,3,4
780 DATA 202,142,1,3,232,240,60,134
785 DATA 12,132,13,169,128,133,27,32
790 DATA 3,4,160,2,177,59,153,0
795 DATA 3,136,16,248,41,255,240,200
800 DATA 48,8,70,27,200,152,145,59
805 DATA 240,190,41,15,201,4,240,5
810 DATA 169,0,170,240,8,160,19,177
815 DATA 59,170,200,177,59,142,3,3
820 DATA 141,4,3,169,255,162,4,76
825 DATA 173,5,120,160,140,185,35,5
830 DATA 153,3,7,136,208,247,88,169
835 DATA 144,133,4,165,4,48,252,72
840 DATA 32,66,208,104,201,1,240,5
845 DATA 162,4,76,65,214,96,35,50
850 DATA 0,32,50,32,80,45,66,36
855 DATA 31,25,18,17,18,19,21,1
860 DATA 2,4,8,16,32,64,128,13
865 DATA 14,8,147,13,13,13,13,32
870 DATA 32,32,32,32,32,32,32,32
875 DATA 32,32,32,198,65,83,84,45
880 DATA 214,65,76,73,68,65,84,69
885 DATA 13,32,32,32,32,32,32,32
890 DATA 32,32,32,32,32,163,163,163
895 DATA 163,163,163,163,163,163,163,163
900 DATA 163,163,13,13,32,32,32,32
905 DATA 32,80,76,85,83,52,32,38
910 DATA 32,214,195,49,53,52,49,32
915 DATA 86,65,76,84,79,90,65,84
920 DATA 58,49,53,83,13,13,32,32
925 DATA 32,32,32,32,213,74,82,65
930 DATA 73,78,68,73,84,65,83,72
935 DATA 79,90,58,210,213,206,32,86
940 DATA 65,71,89,32,198,54,33,13
945 DATA 13,13,32,32,32,32,32,18
950 DATA 32,213,83,83,32,76,69,32
955 DATA 69,71,89,32,66,73,76,76
960 DATA 69,78,84,89,85,84,32,33
965 DATA 32,46,46,46,146,13,13,13
970 DATA 0,32,32,32,32,32,32,32
975 DATA 32,32,32,32,32,32,5,214
980 DATA 193,204,201,196,193,204,207,203
985 DATA 46,46,46,158,13,13,13,0
990 DATA 0,0,66,82,69,65,75,0
995 DATA 80,0,0,0,0,0,0,-1
1000 H=4096:I=.
1005 DO:READA:IFA<.THENEXIT
1010 POKEH,A:H=H+1:I=I+A:LOOP
1015 IFI<>147134THENPRINT"SAJNOS ELRONTOTTAD
     :=> SOK SIKERT ...":END
1020 POKE43,1:POKE44,16:POKE45,92:POKE46,21:END

READY.
```

Hardver kie

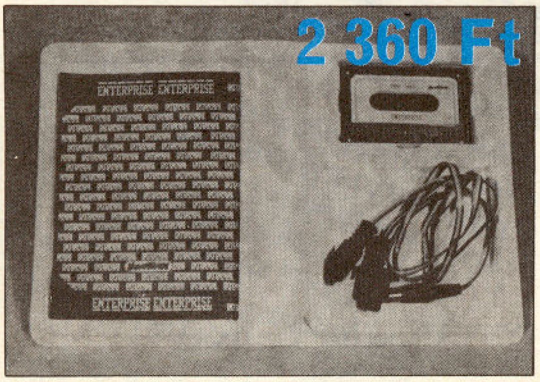
1. Enterprise 128 KByte-os számítógép és magnó



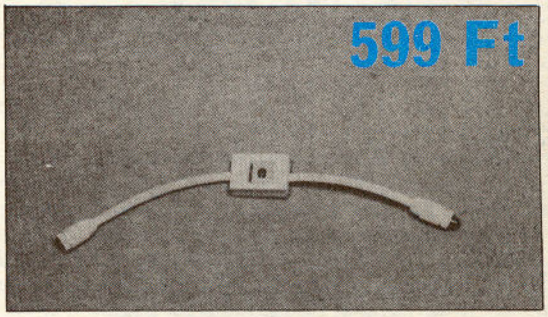
2. Enterprise Euroscart monitorkábel



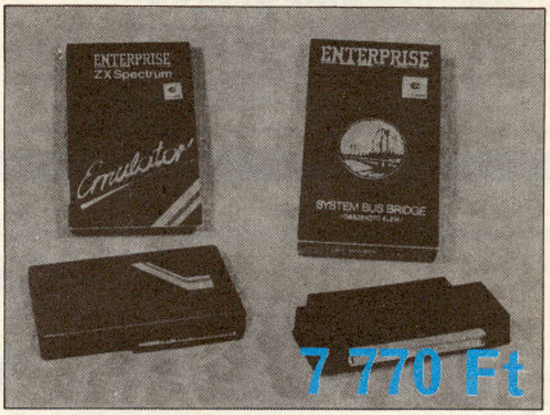
3. Commodore Multi File Transfer



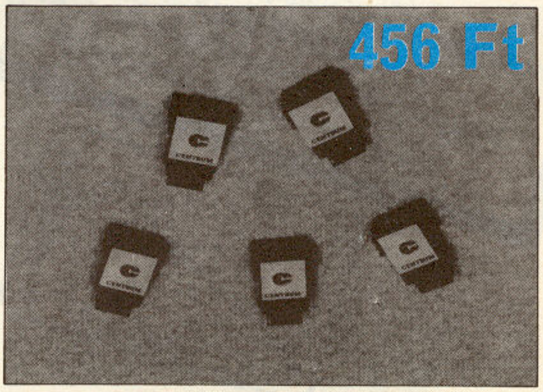
4. Enterprise impedancia-illesztő



5. Enterprise ZX Emulátor és SYSTEM BUS BRIDGE



6. Enterprise Joy-illesztő



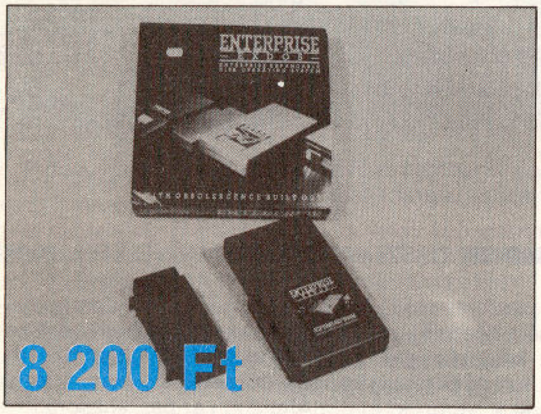
gészítők

A NOVOTRADE RT.
2C Áruházában
megvásárolhatók

7. Enterprise alapgép és
magnó porvédő



8. Enterprise EX—DOS illesztőkártya



9. Enterprise Mouse



10. Enterprise Speak EASY



11. Enterprise DIN monitorkábel



Számítástechnikánk a jövő technikája

Örömmel látjuk a

2c

áruházban

1136 Budapest XIII., Balzac u. 35. Tel. 402-954
Nyitva: hétfőtől péntekig 9-től 18 óráig

Operációk BASIC-ben

Így számol a komputerünk

Állj, állj, szó sincsen semmiféle sebészeti beavatkozásról! Sokkal inkább az érdekel minket, hogyan gondolkodik a C64, a C128 és a C16/Plus4, s hogy ezt hogyan tudjuk kihasználni.

Ezúttal sem sebészeti tárgyú előadást, sem pedig egy hardveres tanfolyamot akarunk tartani. Az a célunk, hogy kitárgyalva a Commodore házi számítógépek által BASIC-ben használatos műveleteket, a kezdő programozónak lehetővé tegyük gépének jobb megértését.

BASIC tanfolyam igen sok van. Ami azonban mindig hiányzik, az az adott parancsok trükkös alkalmazásának ismertetése. Éppen ezért most a matematikai és a logikai műveletekkel fogunk foglalkozni. Ezek a megfelelő alkalmazásnál ugyanis rengeteg időt és tárolót képesek megspórolni nekünk. Egyelőre azonban „hiányoznak" nekünk az alapismeretek, ezért most ezzel kezdjük a tanulást.

Hogy a BASIC-ben végzett műveleteket igazán megérthessük, segítségünkre van a fogalom magyarázata:

Matematikai műveletek

Számítógépes (és matematikai) műveletek alatt értünk mindenféle számítási műveletet, ahol egy matematikai kifejezést a matematika törvényszerűségeit figyelembe véve értékelünk ki és alakítunk át.

Azokat az értékeket, amelyekkel történik valami, operandusnak nevezzük, az operátorok pedig a műveleti előírást adják meg, azaz azt, mi történjen tényelegesen. A mellékelt példán szemléltetjük az elmondottakat.

Az elvégzendő művelet: ÖSSZEADÁS		
3	+	5
1. operandus	operátor (a műveleti előírás)	2. operandus

A C64, a C16/plus4 és a C128 háromféle műveletet (operátort) bocsát a programozó rendelkezésére:

— A számítási műveleteket,
— az összehasonlító műveleteket és a
— logikai műveleteket.

Nézzük meg most az egyes műveleti fajtákat.

A SZÁMÍTÁSI MŰVELETEK

A számítási műveleteket a BASIC-ben a legkülönfélébb számítások elvégzésére használjuk (például W=GXP/100). Ezeket az aritmetikai műveleteket számítási operátorokkal hajtjuk végre. Azt kell itt tudnunk, hogy az operátor mindig az operandusok között áll. Ötféle számítási műveletet ismer a BASIC:

1. Az összeadás (szimbóluma: +)

Az operandusokat összeadjuk. Adjuk be például parancsmódban:

PRINT 1+12+3 ‹RETURN›

A képernyőn a 16-os értéket kapjuk eredményül. Figyelem! A „+" jel használható sztringek „összeadására" (összefűzésére) is, ebben az esetben azonban egy sztringműveletről van szó.

2. A kivonás (szimbóluma: —).

A jobboldalon álló értéket kivonjuk a baloldaliból:

PRINT 9—20 ‹RETURN›

A képernyőn a —‹1 fog áll ezután. Figyelem! Amint azt látjuk, a mínuszjel jelenthet negatív előjelet is.

3. A szorzás (szimbóluma: ×).

Az operandusokat összeszorozzuk.

PRINT 2×5×6 ‹RETURN›

Az alábbi példa eredménye 60 lesz.

4. Az osztás (szimbóluma: /).

A matematika szabálya szerint az osztandó (baloldali érték) elosztjuk az osztóval (jobboldali érték).

PRINT 1/19 ‹RETURN›

Az osztás eredménye 0,052631579.

5. A hatványozás (szimbóluma: ↑).

Az alapot (baloldali érték) a kitevőre (jobboldai érték) emeljük, azaz annyiszor szorozzuk meg önmagával, amennyi a kitevő értéke:

PRINT 3↑3

A művelet eredménye 27 (=3×3×3) lesz.

Ne feledjük el, hogy minden számítási műveletet lebegőpontos számábrázolással (azaz tizedespontos számokkal) hajtunk végre. Amikor megadjuk az értékeket (operandusok), akkor a C64 interpretere átalakítja azokat lebegőpontos alakra, majd a művelet végrehajtása után az eredményt visszaalakítja. Ha a lehetséges maximális értéket túllépjük (+1.70141183 E+38), túlcsordulás hibajelzést kapunk: ?OWERFLOW ERROS. Ha a legkisebb lehetséges pozitív értéken akarunk túllépni (+2.9387588 E—39), eredménynek nullát kapunk.

Ha egy integerként deklarált, azaz egy egész szám esetében lépünk túl a megengedett határokon (—32768‹=x‹=+32767), akkor az ?ILLEGAL QUANTITY ERROR hibajelzéssel kell majd szembenézni. Ha valamikor nullával osztanánk, akkor a ?DIVISION BY ZERO ERROR hibajelzést fogjuk kapni.

Ha a műveleteket kombináljuk egymással, akkor ügyelni kell a prioritásokra. Például a 2+3×2² esetében előbb a hatványozást hajtjuk végre (2²=4), mert ez rendelkezik a legmagasabb prioritással, majd ezt a kapott értéket szorozzuk meg 3-mal (3×4=12), végezetül az összeadás jön. A példánkban ugyanis az összeadás a „legértéktelenebb". Azaz végeredménynek 2+12=14-et kapunk.

a legmagasabb prioritás	↑	hatványozás
	× /	szorzás, osztás
↑	+ —	összeadás, kivonás
↑	‹ = ›	összehasonlító műveletek
↑	NOT	logikai tagadás
a legalacsonyabb	AND	logikai és
prioritás	OR	logikai vagy

Azonos prioritású műveletek esetében balról jobbra történik a műveletek végrehajtása.

ÖSSZEHASONLÍTÁSI MŰVELETEK

Az összehasonlítási műveleteket főleg két operandus összevetésére használjuk. Ez a fajta művelet igaz/hamis értékelést ad. Ha a kifejezés igaz (helyes), akkor „—1" lesz az eredmény, ha az hamis, akkor pedig nulla. (Közelebbit a magyarázó részben a „Logikai változók" alatt.) Általában az összehasonlító műveleteket az IF utasításban használjuk. Ott a kiértékelés eredménye a programozó számára érdektelen.

Hat különböző összehasonlító operátor létezik:

1. › nagyobb mint,
2. = egyenlő,

(Figyelem, az egyenlőségjelet értékadásként is használjuk: pl. A=10, B=C, A$=„MINTA", stb.).

3. ‹ kisebb mint,
4. › = nagyobb egyenlő,
5. ‹ › nem egyenlő,
6. ‹ = kisebb egyenlő.

Például:

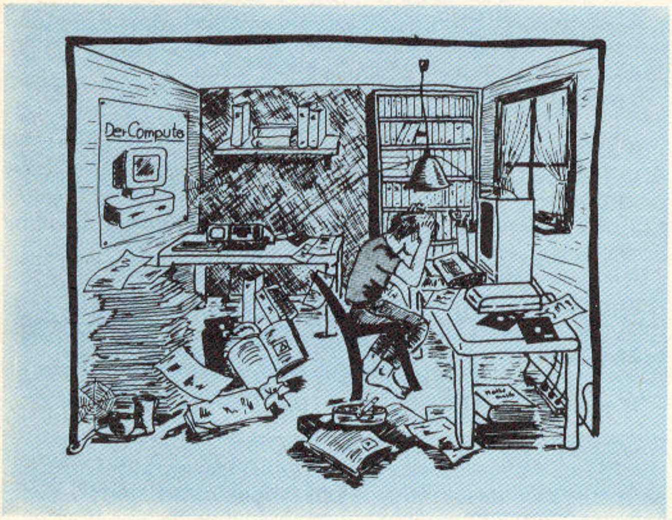
PRINT 3=5 eredménye 0, mivel az állítás hamis.
PRINT 3+4‹7+4 eredménye —1, mivel az állítás igaz.
PRINT 14‹ ›2 eredménye —1, mivel az állítás igaz.

Az összehasonlító műveletekkel nemcsak számokat lehet igaz/hamis alapon letesztelni, hanem sztringeket (karakterláncokat) is. Itt a szringeket karakterenként vetjük össze egymással. Az értékelés alapja a Commodore jelkészlet szerinti kódsorrend, azaz A‹B‹C stb. (Lásd ehhez a gépkönyvet is)

Ebben az összehasonlításban a sztringeknek nem kell azonos hosszúnak lenniük.

Ügyeljünk feltétlenül az összehasonlítási műveletekre. Ha egy numerikus értéket és egy karakterláncot hasonlítunk össze, akkor a ?TYPE MISMATCH ERROR hibajelzést kapjuk.

Mint a számítási műveletek esetében, a BASIC interpreter az összehasonlítási műveleteknél is lebegőpontos számmá alakítja az esetleges egész (integer) számokat.



Azt is jó, ha tudjuk, hogy a gép a hatványozást igen bonyolult módon, a kettes számrendszerben végzi (még a számos alakítgatás után), ennek pedig pontatlanság lesz az eredménye. Ne csodálkozzunk tehát, ha a ?3 3=27 a gép szerint HAMIS állítás!

Az összehasonlító műveleteket gyakran számítják a logikai műveletekhez, mivel ezeknél „igazságérték"-et számolunk ki.

LOGIKAI MŰVELETEK

A logikai műveleteket főleg az összehasonlítási műveletekkel együtt (az IF utasításban) igaz/hamis kiértékelésre használjuk. De a logikai műveletek igazából bináris jegyekkel történő műveleteket jelentenek. A bináris számokról és jegyekről a magyarázó részben olvashatunk még.

A logikai műveleteknél arra kell ügyelni, hogy az operandusok az integer számok tartományába (—32768-tól +32767) essenek, különben ?ILLEGAL QUANTITY ERROR-nak örülhetünk.

Az esetleg előforduló real típusú (lebegőpontos) számokat a művelet során egész számmá alakítjuk, s az eredmény mindig egész szám lesz. A következőkben sorra bemutatjuk majd a logikai műveleteket és azok értelmezését a bit/bit összehasonlításokban. A magyarázó részben részletesen kielemezzük azokat az igazságteszt szempontjából.

A következőkben használni fogunk új fogalmakat is: A biteredmény az lesz, ami az operandusok azonos bithelyeinek összehasonlításából adódik. (Pl. az első operandus negyedik bitjét a második operandus negyedik bitjével. A biteredmény azután a nyert érték negyedik bitje lesz.) Az operandusok a számításra átadott értékek. A logikai műveletek során mindig a decimális számértékek bitmintáival számolunk. A „normál" számokból tehát az eredményt nem lehet „megjósolni". A példák megértéséhez a számokat át kell alakítani bitekké és úgy elvégezni a műveletet, majd az eredményt vissza kell alakítani decimálissá.

BASIC-ben négy logikai műveletet ismerünk:

1. AND (logikai ÉS)

Például ?23 AND 45. (Az eredmény 5 lesz, ellenőrizzük le.)

A logikai ÉS esetében a biteredmény akkor és csak akkor lesz 1, ha az operandusok megfelelő bithelyein álló jegyek mindegyike 1. Ellenkező esetben a biteredmény nulla.

2. OR (logikai VAGY)

Például ? 13 OR 54. (Az eredmény 63 lesz.)

A VAGY logikai művelet esetében a biteredmény akkor és csak akkor lesz 1, ha az operandusok megfelelő bithelyein álló jegyek közül legalább az egyik értéke 1. Ha mindkét bithely értéke nulla, akkor a biteredmény is nulla lesz.

3. NOT (logikai NEM)

Például ? NOT 24. (Az eredmény —25 lesz.)

A NEM utasításnál — itt csak egy operandust adunk át — minden egyes bináris jegyet invertálunk. Nullából egy lesz, 1-ből pedig nulla.

4. XOR (kizáró logikai VAGY) (Csak a C128-asnál!)

Például a BASIC 7.0-ban? XÍR(33,15). (Az eredmény 46 lesz.) Az „exkluzív", vagyis a kizáró VAGY esetében a biteredmény akkor és csak akkor lesz 1, ha az operandusok megfelelő bithelyein ellentétes értékek állnak.

— AND konjukció, egyesítés vagy logikai szorzat 1 AND 1 eredménye 1 0 AND 1 eredménye 0 1 AND 0 eredménye 0 0 AND 0 eredménye 0	— OR diszjunkció, válogatás vagy logikai összeadás 1 OR 1 eredménye 1 0 OR 1 eredménye 1 1 OR 0 eredménye 1 0 OR 0 eredménye 0
— NOT negáció, tagadás vagy logikai invertálás NOT 1 eredménye 0 NOT 0 eredménye 1	— XOR „exkluzív" (kizáró) VAGY vagy antivalencia 1 XOR 1 eredménye 0 0 XOR 1 eredménye 1 1 XOR 0 eredménye 1 0 XOR 0 eredménye 0

A logikai műveletek „igazságtáblázata". A számítógép esetében a logikai műveletek alapjában véve csak a 0 és az 1-es operandusra vonatkoznak. A táblázat jó képet ad az összekapcsolások menetére és szabályára.

A negyedik operátor (XOR) csupán a C128-as BASIC 7.0-ban van külön definiálva. Ügyeljünk a helyes írásmódra. A XOR-nál az operandusokat zárójelben, vesszővel elválasztva kell megadni. Egyébként a C16/Plus4-es és a C64-es is képes a kizáró VAGY leképezésére, ha kihasználjuk a logikai műveletek egymásba alakítási szabályait: XOR (X, Y) = (X OR Y) — (X AND Y).

A mellékelt példákon jól szemlélhetjük a fönti állítások igazát, illetve a logikai műveletek „munkamódszerét" 8 bit műveletek esetében. (A számítógép 2×8=16 bittel számol.)

33	%00100001	Mindig csak a megfelelő bithelyeket vetjük össze. Ha mindkettő "1", úgy a biteredmény is "1" lesz.
AND		
15	%00001111	
1	%00000001	
33	%00100001	A megfelelő bithelyeket összevetjük. A biteredmény akkor lesz "1", ha legalább az egyik bit "1".
OR		
15	%00001111	
47	%00101111	
NOT		Csak egy operandust adunk át. Az eredményben az "1"-ből "0", a "0"-ból "1" lesz. A legszélső (7.) bitet előjelindikátornak használjuk. Ha az "1", úgy negatív, ha "0" akkor pozitív számmal van dolgunk.
33	%00100001	
—34	%11011110	
33	%00100001	
XOR		
15	%00001111	Az egyes bithelyeket összevetjük. A biteredmény csak akkor "1", ha az operandus bithalyek értékei különböznek.
46	%00101110	

Azokat a kifejezéseket, amelyek több, különböző operátort tartalmaznak, egyes részenként dolgozzuk föl. Legelőbb a legmagasabb prioritású, azután az alacsonyabbrendű műveleteket hajtjuk végre. A számítás sorrendjét megváltoztathatjuk (a gömbölyű) zárójelek beszúrásával. Több zárójel is állhat a kifejezésben (de mindig párjával — "(" és ")" — kell szerepeltetni őket, különben ?SYNTAX ERROR hibajelzést kapunk). A zárójeleket dobozolni is lehet egymásba, de maximálisan tíz síkot (zárójelpárt) szerepeltethetünk. Ekkor előbb a legbelsőt bontja föl az interpreter. Azonos prioritások esetében, mint az már mondtuk, a kifejezés kiértékelése balról jobbra történik.

A legmagasabb prioritással a számítási operátorok rendelkeznek (saját fokozatokban), ezt az összehasonlító operátorok követik. A logikaiaké a legalacsonyabb prioritás (itt is van egy sorrend). Például a ?3×4+1 eredménye 13 lesz, a ?3×(4+1) a 15-öt adja. Föntebb megnézhetjük az elsőbbségi sorrendet.

FOGALOMMAGYARÁZAT

1. Logikai változók:

Amint azt láthattuk, a számítógép egy igaz kifejezés kiértékelésekor a "—1" értéket adja, hamisnál pedig a "0"-t. Tulajdonképpen létezik egy változótípus az igaz/hamis kiértékeléshez. Ezeket logikai vagy boolean változóknak nevezzük. Ezek csak az „igaz" és a „hamis" értékeket képesek fölvenni.

A Commodore kisgépek esetében ezt a változótípust nem különítik el más, numerikus változóktól (azaz erre a célra nincs külön változódefiníció). Emiatt azután azokkal a változókkal, amelyekhez logikai igazságértéket rendeltünk hozzá, később más műveleteket is végezhetünk.

2. A bináris (duális) rendszer:

Az általunk ismert és használt tizesalapú számrendszerrel (tíz jegy) szemben a számítógép kettes alapon dolgozik. Itt csak két jegy, a nulla és az egyes áll a rendelkezésre.

Mialatt a tizes rendszer számainál a tizedesponttól jobbra álló számjegyek értéke a tíz hatványaival (1, 10, 100, 1000, stb.) növekszik, a bináris rendszerben ez a faktor a 2 (1, 2, 4, 8, 16, . . .).

Egy n-jegyű szám alakjai a két számrendszerben:

tizes: $a_n \times 10^n + a_{n-1} \times 10^{n-1} + \ldots + a_2 \times 10^2 + a_1 \times 10 + a$

kettes: $a_n \times 2^n + a_{n-1} \times 2^{n-1} + \ldots + a_2 \times 2^2 + a_1 \times 2 + a$

A kettes számrendszer jegyeit a számítógép jól tudja „értelmezni", hisz ez a be/ki, a van áram/nincs áram, állapotot képezi le. Decimális számot az alábbi módon alakíthatunk át:

Például a 221 (= 2×100+2×10+1) binárisan az 1101 1101-nek (= 1×128+1×64+0×32+1×16+1×8+1×4+0×2+1).

Az átszámítási mechanizmus egyszerű:

221/2 = 110, a maradék 1 0. bit
110/2 = 55, a maradék 0 1. bit
55/2 = 27, a maradék 1 2. bit
27/2 = 13, a maradék 1 3. bit
13/2 = 6, a maradék 1 4. bit
6/2 = 3, a maradék 0 5. bit
3/2 = 1, a maradék 1 6. bit
1/2 = 0, a maradék 1 7. bit = %1101 1101

A 221 bináris megfelelője tehát az 1101 1101 let. Addig kellett tehát osztani kettővel, amíg a decimális szám el nem „tűnt". A bináris számot a fönti séma szerint azután alulról kezdve kell „összeolvasni", mint ahogy a decimális rendszerben is a legnagyobb helyiértékű szám áll legelál. A bináris számjegyek „értékei" mindig az adott osztás maradékai lesznek. Hogy a bináris számokat meg lehessen különböztetni másoktól, ezért a szám elé a százalékjelet (%) írjuk: %221 = 1101 1101.

3. Logikai műveletek és az igaz/hamis kiértékelése.

A számítógép egy igaz kifejezés értékének „—1"-et, hamisénak nullát ad meg. Ha a két érték bináris alakját nézzük, megállapíthatjuk, hogy a —1 ellentéte épp a nulla:

0 = 0000 0000 0000 0000
—1 = 1111 1111 1111 1111

A számítógép az egész számokat 16 biten (0—15) ábrázolja. A 15. bitet azonban félre kell tennünk az előjel számára. Ha a 15. bit értéke egy, úgy negatív, ha nulla, akkor pozitív számról van szó. Nullánál minden bit nulla, a —1 esetében minden bit értéke egy.

A negatív számokat különleges módszer szerint kell ábrázolni. Ha látjuk, hogy a 15. bit egy, leválasztjuk azt. A maradék jegyeket NOT paranccsal átbillentjük, majd a kapott eredményhez hozzáadunk 1-et:

—1 = 1111 1111 1111 1111 = › 1/ 000 0000 0000 0000 +1 = ›—1
—34 = 1111 1111 1111 1101 = › 1/ 000 0000 0010 0001 + 1 = = ›—34

(Lásd a NOT 33 példát föntebb.)

A logikai változók csak egy bitet tartalmaznak. Mivel a BASIC az összehasonlításoknál és a logikai igaz/hamis kiértékelésnél az integer értéktartományt használja, definiálni kellett az egyértelmű állapotokat:

Minden bit bekapcsolva = igaz. Minden bit kikapcsolva = hamis. A definíció korrekt, a logikai műveletek a megfelelő értékeket adják (lásd föntebb az igazságtáblázatot):

—1 AND —1 = —1 —1 OR —1 = —1 —1 XOR —1 = 0
XOR (—1, —1) = 0
0 AND —1 = 0 0 OR —1 = —1 0 XOR —1
XOR (0, —1) = —1
—1 AND 0 = 0 —1 OR 0 = —1 —1 XOR 0 = —1
XOR (—1, 0) = —1
0 AND 0 = 0 0 OR 0 = 0 0 XOR 0 = 0
XOR (0, 0) = 0

Az XOR csak a BASIC 7.0-ban van parancsként definiálva, adott esetben azt a föntebb megadott módon kell megvalósítani. Az igaz/hamis kiértékelést leggyakrabban az IF utasításnál használjuk:

10 IF A = 3 And Y = 13 THEN 30

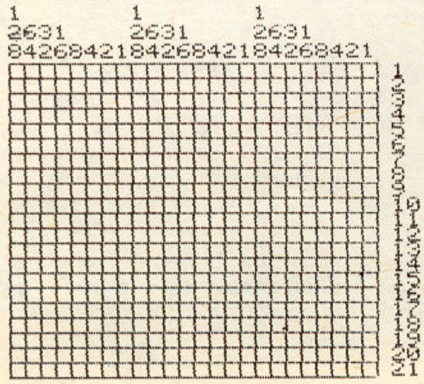
Ebben az esetben kombináljuk az összehasonlítási és a logikai műveleteket az igazság kiértékeléséhez. (Akkor léphetünk csak a 30. sorra, ha az A = 3 és az Y = 13 feltétel egyszerre teljesül.

Figyelem! Az IF utasítás egy kifejezést akkor ismer el igaznak, ha az nullától különböző értékű. Azaz akkor is, ha a kifejezés értéke nem —1, de nem nulla. Az IF tehát minden nulla értéket adó kifejezést hamisnak talál.

PROGRAM

Sprite tervező segédlet

Az alábbi kis program sprite-ok tervezéséhez nyújt segítséget. Hatására a nyomtatón a 64-es kézikönyvéből megismert sprite tervező rács jelenik meg.



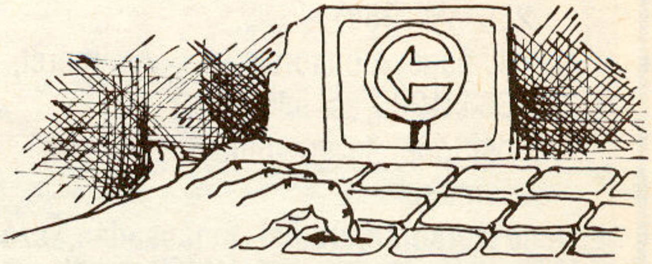

```
1 REM ******************************
2 REM *     C= UJSAG  SORSZAM:162  *
3 REM *     SPRITE TERVEZO         *
4 REM *                            *
5 REM ******************************
100 PRINT"SPRITE-TERVEZO NEGYZETRACS"
110 PRINT"   'P' = PRINT VAGY"
120 PRINT"   'V' = VEGE .."
130 GET A$
140 IF A$="V" THEN END
150 IF A$<>"P" THEN 130
160 A$=CHR$(15):B$=CHR$(8):OPEN4,4:CMD4
170 FOR J=1 TO 3:PRINT"1";SPC(7);:NEXT:PRINT
180 FOR J=1 TO 3:PRINT"2631";SPC(4):NEXT:PRINT
190 FOR J=1 TO 3:PRINT"84268421";:NEXT:PRINT
200 FOR R=1 TO 21:FOR C=1 TO 24:PRINT CHR$(111);:NEXT
210 PRINT CHR$(165);MID$(STR$(R),2);B$:PRINT A$;:NEXT
220 FOR J=1 TO 24:PRINT CHR$(163);:NEXT
230 PRINT#4:CLOSE4

READY.
```

Balnyílcopy

BASIC nyelvű programok dokumentálásához nagy segítséget jelentene, ha a program egyes részeiről, pl. menükről, bejelentkezésekről képernyőmásolatot tudnánk készíteni. Ez általában nem jelent különösebb gondot, mivel a szaklapok hasábjain már megjelentek hardcopy szubrutinok, vagy gépi kódú sys-sel indítható hardcopy-k. Ezek használata egy kicsit kényelmetlen. Az ideális az lenne, ha a képernyőről bármikor tudnánk készíteni másolatot, egyszerűen egy billentyű lenyomásával. A bemutatásra kerülő program éppen erre a célra készült. Minden olyan BASIC nyelvű program képernyőtartalmáról készíthetünk vele papírra másolatot, amely nem használja a C000-ás és a 033c területet és nem módosítja az IRQ vektort.

Egy ilyen program megvalósítását a C64 operációs rendszerének nyitottsága teszi lehetővé. Ami alatt azt kell értenünk, hogy kívülről könnyen hozzáférhető és módosítható. Az IRQ vektor a 0314 és 0315-ös címeken helyezkedik el a RAM-ban, és az EA31 ROM-ban elhelyezkedő megszakítás rutin címére mutat. Az IRQ vektort egy saját rutinra állítjuk, amely a balnyíl billentyűt kérdezi le. Ha ez a rutin billentyű lenyomást észlelt, akkor egy hardcopy rutinra ugrik, majd az IRQ rutinra. Ha nem volt balranyíl billentyű lenyomás, akkor az IRQ rutinra ugrik. A hardcopy rutinnál le kell tiltani a megszakítást és gondoskodni kell a balranyíl kód képernyőtárból és az átmeneti tárból történő törléséről. Mivel a nyomtatók részére ASCII kódot kell küldeni, ezért a képernyőkódot először át kell ASCII kódra alakítani.



A balnyílcopy assembler forrásprogramja az 1-es listán tanulmányozható és profi assembler segítségével a C64 tárába illetve lemezre vihető. Aki ilyennel nem rendelkezik, az a 2-es lista BASIC betöltőjét a c000-ás, a 3-as listáét a kazettapuffer 033c területre történő betöltéshez használhatja. A használattal kapcsolatban ne feledjük el, hogy először a balnyílcopy programot kell betölteni és utána a másolandót.

Horváth Józsefné

```
10 OPEN1,8,1,"F BALNYILCOPY"
20 OPEN2,4
30 SYS9*4096
40 .OPT OO,P
100 ; NYOMTATAS BALRA NYILRA
110 ;
120 ;
130 ; A KARAKTERES KEPERNYO NYOMTATASA
140 ; HIVAS     SYS 12*4096
150 ;
151 IRQVEC = $0314
152 IRQRUT = $EA31
160 TEMP   = $FC
170 PT     = $FD
180 SCREEN = $0400
190 SETFPA = $FFBA
200 SETFNA = $FFBD
210 OPEN   = $FFC0
220 CKOUT  = $FFC9
230 CLRCH  = $FFCC
240 BSOUT  = $FFD2
250 COLUMN = 40
260 ROW    = 25
270 CR     = 13
280 *=$C000
290 AKTIV SEI; AZ IRQVEKTOR ATIRASA
300 LDA #<NYIL
310 STA IRQVEC
320 LDA #>NYIL
330 STA IRQVEC+1
340 CLI
350 RTS
360 ; AZ UJ IRQ RUTIN
370 NYIL LDA $CB; LENYOMOTT BILLENTYU
380 CMP #$39
390 BEQ HARD
400 LI JMP IRQRUT
401 ;
402 HARD SEI
403 LDA #$00
404 STA $CB
407 LDA #$20     ;A CURSOR BALRA
408 DEC $D3      ;MOZGATASA ES
409 LDY $D3      ;A BALNYIL KOD
410 STA ($D1),Y  ;KITORLESE A TARBOL
420 LDA #127     ; OPEN 127,4,0,
430 LDX #4
440 LDY #0
450 JSR SETFPA
460 LDA #0
470 JSR SETFNA
480 JSR OPEN
490 BCS QUIT
500 LDA #<SCREEN   ; A KEPERNYO KEZDETE
```

```
510 STA PT
520 LDA #>SCREEN
530 STA PT+1
540 LDX #127          ; PRINT#127,...
550 JSR CKOUT
560 LDX #ROW          ; A SOROK SZAMA
570 ROWLOO LDA #CR          ; NYOMTATO CIKLUS
580 JSR BSOUT
590 JSR $FFE1         ; STOP-GOMB FIGYELES
600 BEQ QUIT
610 LDY #0
620 COLLOO LDA (PT),Y
630 STA TEMP          ; KEPERNYOKOD->ASCII
640 AND #$3F
650 ASL TEMP
660 BIT TEMP
670 BPL NOOR
680 ORA #$80
690 NOOR BVS NOADDO
700 ORA #$40
710 NOADDO JSR BSOUT
720 INY
730 CPY #COLUMN
740 BNE COLLOO
750 TYA
760 CLC
770 ADC PT
780 STA PT
790 BCC NOOR1
800 INC PT+1
810 NOOR1 DEX
820 BNE ROWLOO
830 LDA #CR          ;"KESZ?"
840 JSR BSOUT
850 JSR CLRCH
860 QUIT LDA #127      ; CLOSE 127
870 JSR $FFC3      ; CLOSE
875 CLI
880 JMP LI
890 .END

READY.
```

```
10 REM***************************
20 REM*       BALNYILCOPY       *
30 REM*        ,  ,             *
40 REM*    HORVATH JOZSEFNE     *
50 REM*                         *
60 REM*         1989            *
70 REM*       BUDAPEST          *
80 REM*                         *
90 REM***************************
100 FOR I =828TO965:READQ:S=S+Q:POKEI,Q:NEXT
101 DATA 120,169,73,141,20,3,169,3,141,21,3,88,96,165,203,201
102 DATA 57,240,3,76,49,234,120,169,0,133,203,169,32,198,211,164
103 DATA 211,145,209,169,127,162,4,160,0,32,186,255,169,0,32,189
104 DATA 255,32,192,255,176,74,169,0,133,253,169,4,133,254,162,127
105 DATA 32,201,255,162,25,169,13,32,210,255,32,225,255,240,49,160
106 DATA 0,177,253,133,252,41,63,6,252,36,252,16,2,9,128,112
107 DATA 2,9,64,32,210,255,200,192,40,208,230,152,24,101,253,133
108 DATA 253,144,2,230,254,202,208,205,169,13,32,210,255,32,204,255
109 DATA 169,127,32,195,255,88,76,79,3,0
110 IFS<>17956THENPRINT"HIBA A DATASORBAN":END
120 SYS828:PRINT"OK!!"

READY.
```

```
10 REM***************************
20 REM*       BALNYILCOPY       *
30 REM*        ,  ,  ,          *
40 REM*    HORVATH JOZSEFNE     *
50 REM*                         *
60 REM*         1989            *
70 REM*       BUDAPEST          *
80 REM*                         *
90 REM***************************
100 FOR I =49152TO49289:S=S+Q:READQ:POKEI,Q:NEXT
101 DATA 120,169,13,141,20,3,169,192,141,21,3,88,96,165,203,201
102 DATA 57,240,3,76,49,234,120,169,0,133,203,169,32,198,211,164
103 DATA 211,145,209,169,127,162,4,160,0,32,186,255,169,0,32,189
104 DATA 255,32,192,255,176,74,169,0,133,253,169,4,133,254,162,127
105 DATA 32,201,255,162,25,169,13,32,210,255,32,225,255,240,49,160
106 DATA 0,177,253,133,252,41,63,6,252,36,252,16,2,9,128,112
107 DATA 2,9,64,32,210,255,200,192,40,208,230,152,24,101,253,133
108 DATA 253,144,2,230,254,202,208,205,169,13,32,210,255,32,204,255
109 DATA 169,127,32,195,255,88,76,19,192,255
110 IFS<>18214THENPRINT"HIBA A DATASORBAN":END
120 SYS49152:PRINT"OK!!"

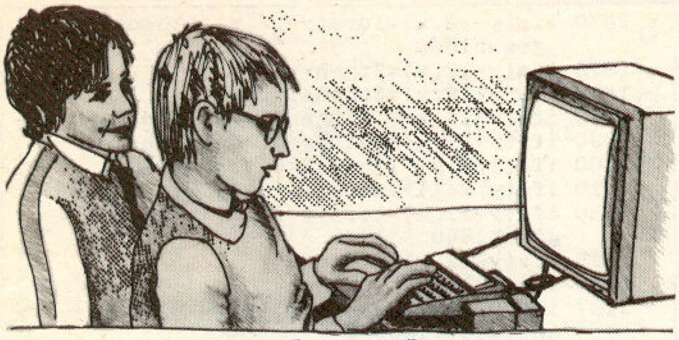
READY.
```

hírek

Számítógépes álomország

A barkácsolók és technikabarátok álomországaként mutatkozott be 1988. 11. 10. és 13. között a Hobby + Elektronik kiállítás Stuttgartban. A szervezők elmondása szerint a látogatók száma 75 000 volt, ami egyértelműen nagyobb, mint az előző évben. A kiállítók száma 343 volt, ez 45%-kal több, mint tavaly. A számos játék és egy nagy modellkiállítás mellett sok számítógépet és tartozékokat láthattunk. Néhány különösen alacsony ár igencsak meglepte a közönséget. Például egy vadonat új, eredeti csomagolású Atari XL 600-ast 20 (írd és mondd húsz) márkáért lehetett venni, amíg a készlet tartott. 300 márkáért Sinclair Spectrum II-t adtak 100 játékkal együtt. A Horten cég kiállítási standján Atari XL, C16-os, C64-es játékkazettákat árultak az alábbi felirattal: „Használhatók üres kazettaként is!" Egy ilyenért 1,95 márkát kértek. Megjegyzendő, volt egy raklap Datasette, melynek darabját 19,50-ért vihette, aki akarta, és színes monitor (garancia nélküli 1901-es 250 márkáért). A kiállítással egyidőben zajlott a Holographics is, ahol az érdeklődők mintegy 2000 négyzetméteres területen informálódhattak a holografika technikáiról. A következő Hobby + Elektronik 1989. novemberében lesz.

Info.: Stuttgarter Messe- und Kongressgesellschaft mbH, Postfach 990, D—7000 Stuttgart 1.



JÁTÉK

3—D LABIRINTUS

A program az ismert 3-dimenziós labirintusok egyik formája. A megvalósítás hasonló, a feladat azonban egy kicsit összetettebb. A játék kezdetén egy szobákból álló labirintus legkisebb termében vagyunk. A szobákban különböző kincsek, ékszerek (a térkép ‚A' betűvel jelöli őket) vannak elhelyezve. Ezeket kell a játék folyamán összegyűjtenünk. Mivel a vándorlás során megszomjazunk, ezért időnként vizet kell magunkhoz venni (‚V' betűvel van jelölve a térképen.) Ha ezt nem tesszük meg időben, akkor először szomjasak leszünk, majd elhalálozunk (sajnos a víz csak korlátolt mennyiségben található a teremben.) Ha a szobában az összes kincset magunkhoz vettük, akkor menjünk ki a kijáraton (ez az ajtó csak abban az esetben nyílik ki, ha már minden kincset összegyűjtöttünk.) A következő szobában — melynek mérete nagyobb, mint az őt megelőzőé — a feladat nem változik, a továbbjutási esély mégis kisebb lesz a pálya mérete miatt. A játék célja minél több pont szerzése, valamint a labirintus legnagyobb termeinek az elérése. Ha jól játszottunk, akkor a játék vége után felkerülhetünk a pontszámtáblázatra is. A programban lehetőség nyílik a táblázat kimentésére és visszatöltésére is, de csak lemezre. Játék közben van egy segítségünk is; egy műszerfal, amely tartalmazza irányunkat, a nálunk levő víz mennyiségét, a még felszedendő aranytárgyak számát, a kincs és a víz távolságát tőlünk és azt, hogy hányszor vehetjük még elő zsebünkből a közben egyre jobban elrongyolódott térképet.

Bár a program fő része BASIC nyelven íródott, sebessége mégsem hoz szégyent a 64-esre a kirajzolást segítő gépi kódú rutinok miatt.

A játék a kettes PORT-ba illesztett botkormányról vezérelhető. A program használatához sok sikert kíván a játék szerzője:

DEBRE ATTILA

163

```
0 ifpeek(4*4096)=1thenre=1:poke4*4096,0
1 c=rnd(76):goto18
2 v=20480
3 data65,84,73,83,79,70,84,32,32,32,234,
  96
4 fori=0to11:reada:pokev+i,a:next
5 v=v+i:fori=1to10:fort=1to10
6 pokev+t,32:nextt
7 pokev+11,00:pokev+12,00:v=v+13
8 nexti:return
18 poke820,0:la=0:ifre=1thenfori=0to11:
   reada:next:goto20
19 gosub2
20 poke53280,0:poke53281,0:poke646,14
30 print"[SHF CL/HOME]":z=200:gosub1680:
   poke827,10
40 print"[SHF CL/HOME][CRS LE][CRS LE]
   [CRS LE]":ir=1:m=z/100
70 m=m*2:dima(m,m):ir$="eszak":te=m/2
71 poke53265,27
75 y=m:x=m-1:at=0:vt=0:gosub670
80 fori=0tom:fort=0tom:a(i,t)=4:next:
   next
90 fori=2tom:a(0,i)=1:next
95 s=226:z=200:ir=1
100 fori=0tom:a(i,0)=1:next
110 fori=0tom:a(i,m)=1:next
120 fori=1tom-2:a(m,i)=1:next
130 ifm=2thena(m,1)=5:goto440
135 a(0,1)=5:a(m,m-1)=5
140 fori=2tom-2step2:fort=2tom-2step2
150 a(i,t)=1
160 next:next:x1=m-1:y1=m+1
170 fori=2tom-2step2:fort=2tom-2step2
180 rn=int(rnd(1)*4)+1
190 onrngoto200,220,240,260
200 ifa(i+1,t)=1then180
210 a(i-1,t)=1:goto280
220 ifa(i-1,t)=1thcn180
230 a(i+1,t)=1:goto280
240 ifa(i,t+1)=1then180
250 a(i,t-1)=1:goto280
260 ifa(i,t-1)=1then180
270 a(i,t+1)=1:goto280
280 next:next
290 fori=1tom/2
300 q1=int(rnd(1)*m):q2=int(rnd(1)*m)
310 ifa(q1,q2)<>4then300
320 gosub390:ifc=1then300
330 a(q1,q2)=2:next:c=2:vi=99.99
340 fori=1tom/2
350 q1=int(rnd(1)*m):q2=int(rnd(1)*m)
360 ifa(q1,q2)<>4then350
370 gosub390:ifc=1then350
380 a(q1,q2)=3:next:ar=m/2
382 ifla=1thengosub440:goto880
384 goto730
390 ifq1>m-1orq1<1orq2>m-1orq2<1then430
395 c=0:ifa(q1+1,q2)=4thenreturn
400 ifa(q1-1,q2)=4thenreturn
410 ifa(q1,q2+1)=4thenreturn
420 ifa(q1,q2-1)=4thenreturn
430 c=1:return
440 print"[CL/HOME]":poke646,14
450 ifla=1thend1=6:gosub1640:goto470
460 print"[SHF CL/HOME]"
470 fori=0tom:fort=0tom:b=a(i,t)
480 ifb=1thenprint"[CTRL 9] [CTRL 0]";:
    nextt
490 ifb=4orb=5thenprint" ";:nextt
500 ifb=2thenprint"v";:nextt
510 ifb=3thenprint"a";:nextt
520 print"[CRS LE][CRS FEL]":nexti:ifla=
    1then620
530 print"[CL/HOME][CRS LE][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB]labirintus"
540 print"[CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]------
    --------"
550 print"[CRS LE][CRS LE]":goto570
560 print"[CRS LE][CRS JOBB][CRS JOBB]
    [CRS JOBB][CRS JOBB][CRS JOBB]
```

```
     [CRS JOBB][CRS JOBB][CRS JOBB]
     [CRS JOBB][CRS JOBB][CRS JOBB]
     [CRS JOBB][CRS JOBB][CRS JOBB]
     [CRS JOBB][CRS JOBB][CRS JOBB]
     [CRS JOBB][CRS JOBB][CRS JOBB]
     [CRS JOBB][CRS JOBB][CRS JOBB]
     [CRS JOBB]";:return
570 gosub560:print"h => segitseg"
580 gosub560:print"irany : ";ir$
590 gosub560:print"v -->      viz"
600 gosub560:print"a -->    arany"
610 gosub560:print"[CRS LE]terkep : t"
620 poke211,x1:poke214,y1:sys58732
630 rem↑↑akurzorkoordinatai
640 poke646,1:poke204,0:remkurzorbe
659 fort=0tom*400:next:poke646,14
660 poke207,0:poke204,1:return
670 l1=1:l2=2:l3=3:l4=4:k=1:return
680 l1=-1:l2=-2:l3=-3:l4=-4:k=-1:return
690 ifir=1thenir$="eszak  ":return
700 ifir=2thenir$="kelet  ":return
710 ifir=3thenir$="del    ":return
720 ifir=4thenir$="nyugat":return
730 gosub440:ti$="000000":la=1:te=m/2
740 gosub670:poke53265,11
750 print"[SHF CL/HOME]";tab(25)"[C= A]C
    CCCCCCCCCCC[C= S]"
760 printtab(25)"B labirintus B"
770 printtab(25)"[C= Z]CCCCCCCCCCCC
    [C= X]"
780 po=0:v1=50:y=m:x=m-1:pt=0:at=0:vt=0
790 printtab(67)"viz"tab(104)"[C= A]CCCC
    C[C= S]"
800 printtab(24)"B[C= J]    [C= L]B"tab(6
    4)"[C= Z]CCCCC[C= X]"
810 print"[CRS LE][CRS LE]"tab(28)"ponts
    zam"
820 printtab(28)"[C= A]CCCCCCC[C= S]"
830 printtab(28)"[CTRL 9][C= K][CTRL 0]
          [C= K]"tab(68)"[C= Z]CCCCCCC
    [C= X]"
840 print"[CL/HOME]"tab(193)"arany"tab(1
    13)"[C= A]CCCC[C= S]"
850 printtab(33)"B[C= J]   [C= L]B"tab(73
    )"[C= Z]CCCC[C= X]"
860 print"[CRS LE][CRS LE][CRS LE]
    [CRS LE][CRS LE][CRS LE][CRS LE]"
    tab(24)"arany tavolsaga"
870 print"[CRS LE][CRS LE][CRS LE]"tab(2
    5)"viz tavolsaga"
871 print"[CRS LE][CRS LE]irany:
    terkep"
880 rem*********
890 poke646,1:poke53265,11:vi=vi-3
895 ify=0andar=0andir=1then1585
896 gosub1610:at=0:vt=0
900 ifir=2orir=4then1140
910 ifir=3then970
920 ify=3then1020
930 ify=2then1050
940 ify=1then1080
950 ify=0andar>0theny=1:goto1080
960 goto1010
970 ify=m-3then1020
980 ify=m-2then1050
990 ify=m-1then1080
1000 ify=mtheny=m-1:goto1080
1010 ifa(y-14,x)=1ora(y-14,x)=5thend1=11
     :gosub1640
1020 ifa(y-13,x-k)>1thend1=9:gosub1640
1030 ifa(y-13,x+k)>1thend1=10:gosub1640
1040 ifa(y-13,x)=1ora(y-13,x)=5thend1=4:
     gosub1640
1045 ifa(y-13,x)=2thenvt=3
1046 ifa(y-13,x)=3thenat=3
1050 ifa(y-12,x-k)>1thend1=2:gosub1640
1060 ifa(y-12,x+k)>1thend1=3:gosub1640
1070 ifa(y-12,x)=1ora(y-12,x)=5thend1=12
     :gosub1640
1075 ifa(y-12,x)=2thenvt=2
1076 ifa(y-12,x)=3thenat=2
1080 ifa(y-11,x-k)>1thend1=7:gosub1640
1090 ifa(y-11,x+k)>1thend1=8:gosub1640
1100 ifa(y,x-k)>1thend1=0:gosub1640
1110 ifa(y,x+k)>1thend1=1:gosub1640
1120 ifa(y-11,x)=1ora(y-11,x)=5thend1=5:
     gosub1640
1125 ifa(y-11,x)=2thenvt=1
1126 ifa(y-11,x)=3thenat=1
1127 ifa(y,x)=2thena(y,x)=4:vt=0:pt=pt+
     int(vi):vi=99.9
1128 ifa(y,x)=3thena(y,x)=4:at=0:pt=pt+z
     :ar=ar-1
1130 goto1322
1140 ifir=2then1190
1150 ifx=3then1220
1160 ifx=2then1250
1170 ifx=1then1280
1180 goto1210
1190 ifx=m-3then1220
1200 ifx=m-2then1250
1205 ifx=m-1then1280
1210 ifa(y,x-14)=1ora(y,x-14)=5thend1=11
     :gosub1640
1220 ifa(y+k,x-13)>1thend1=9:gosub1640
1230 ifa(y-k,x-13)>1thend1=10:gosub1640
1240 ifa(y,x-13)=1ora(y,x-13)=5thend1=4:
     gosub1640
1245 ifa(y,x-13)=2thenvt=3
1246 ifa(y,x-13)=3thenat=3
1250 ifa(y+k,x-12)>1thend1=2:gosub1640
1260 ifa(y-k,x-12)>1thend1=3:gosub1640
1270 ifa(y,x-12)=1ora(y,x-12)=5thend1=12
     :gosub1640
1275 ifa(y,x-12)=2thenvt=2
1276 ifa(y,x-12)=3thenat=2
1280 ifa(y+k,x-11)>1thend1=7:gosub1640
1290 ifa(y-k,x-11)>1thend1=8:gosub1640
1300 ifa(y+k,x)>1thend1=0:gosub1640
1310 ifa(y-k,x)>1thend1=1:gosub1640
1315 ifa(y,x-11)=1ora(y,x-11)=5thend1=5:
     gosub1640
1316 ifa(y,x-11)=2thenvt=1
1317 ifa(y,x-11)=3thenat=1
1318 ifa(y,x)=3thena(y,x)=4:at=0:pt=pt+z
     :ar=ar-1
1320 ifa(y,x)=2thena(y,x)=4:vt=0:pt=pt+
     int(vi):vi=99.9
1321 goto1340
1322 ify=1andx=1andir=1then1324
1323 goto1325
1324 print"[CL/HOME][CRS LE][CRS LE]
     [CRS LE][CRS LE]"tab(207)"kijarat"
1325 ify=m-1andx=m-1andir=3then1327
1326 goto1340
1327 print"[CL/HOME][CRS LE][CRS LE]
     [CRS LE][CRS LE]"tab(207)"bejarat"
1340 poke53265,27
1360 print"[CL/HOME][CRS LE][CRS LE]"
     tab(235)ar
1370 print"[CRS LE][CRS LE][CRS LE]
     [CRS LE][CRS LE]"tab(29)pt
1380 print"[CRS LE][CRS LE][CRS LE]
     [CRS LE]"tab(30)at
1390 print"[CRS LE][CRS LE][CRS LE]"tab(
     30)vt
1392 printtab(6)ir$"[CRS JOBB][CRS JOBB]
     [CRS JOBB][CRS JOBB][CRS JOBB]
     [CRS JOBB][CRS JOBB][CRS JOBB]
     [CRS JOBB]"te
1400 getf$:iff$<>"t"then1430
1410 te=te-1:ifte<0thente=0:goto1430
1420 gosub440:goto890
1430 a1=peek(56320)
```

```
1440 ifal=126then1480
1450 ifal=123then1490
1460 ifal=119then1540
1462 ifvi<10thens=227:print"[CL/HOME]
     [CRS LE][CRS LE]"tab(s)" "
1465 print"[CL/HOME][CRS LE][CRS LE]"
     tab(s)int(vi)
1467 vi=vi-.5:ifvi<.5then5000
1470 goto1400
1480 ifir=1anda(y-1,x)>1theny=y-1:y1=y1-
     1:goto880
1481 ifir=2anda(y,x+1)>1thenx=x+1:x1=x1+
     1:goto880
1482 ifir=3anda(y+1,x)>1theny=y+1:y1=y1+
     1:goto880
1483 ifir=4anda(y,x-1)>1thenx=x-1:x1=x1-
     1:goto880
1484 goto1400
1490 ify=mandir<>3then1430
1500 ir=ir-1:ifir=0thenir=4
1510 gosub690:ifir=1orir=4thengosub670
1520 ifir=2orir=3thengosub680
1530 goto880
1540 ify=mandir<>3then1430
1550 ir=ir+1:ifir=5thenir=1
1560 gosub690:ifir=1orir=4thengosub670
1570 ifir=2orir=3thengosub680
1580 goto880
1585 d1=6:gosub1640:m=m/2+1:ifm=11thenm=
     10
1586 p1=int(pt/256):p2=pt-p1*256
1587 poke821,m:poke822,p1:poke823,p2:clr
1588 m=peek(821):pt=peek(822)*256+peek(8
     23):la=1:g=1
1589 fori=0to11:reada:next:gosub1680
1590 print"[CL/HOME][CRS LE][CRS LE]"
     tab(227)" "
1591 print"[CRS LE][CRS LE][CRS LE]
     [CRS LE][CRS LE][CRS LE][CRS LE]
     [CRS LE][CRS LE][CRS LE][CRS LE]
     [CRS LE][CRS LE][CRS LE][CRS LE]"
     tab(20)" "
1592 goto70
1610 poke49176,0:poke49186,0:poke49193,0
1620 poke49177,196:poke49187,196
1630 poke49194,196:sys49152:return
1640 poke49176,po(d1):poke49186,po(d1)
1650 poke49193,po(d1)
1660 poke49177,197:poke49187,197
1670 poke49194,197:sys49152:return
1680 dimpo(20):fori=0to12:reada
1690 po(i)=a:next:ifg=1thenreturn
1700 ifpeek(820)=16thenprint"
     [SHF CL/HOME]":goto1790
1705 ifre=1thenprint"[SHF CL/HOME]":goto
     1790
1710 printtab(132)"egy kis turelmet !"
1720 poke820,16
1730 data9,19,35,60,88,125,132,139,164
1740 data192,205,221,252
1750 fori=0to175:reada
1760 poke49152+i,a:next
1770 fori=0to580:reada
1780 poke50176+i,a:next:print"
     [SHF CL/HOME]"
1790 poke53265,11:gosub1610
1800 d1=0:gosub1640:d1=1:gosub1640
1810 d1=2:gosub1640:d1=3:gosub1640
1820 d1=7:gosub1640:d1=8:gosub1640
1830 d1=11:gosub1640:poke53265,27
1840 print"[CL/HOME]";tab(25)"[C= A]CCCC
     CCCCCCCC[C= S]"
1850 printtab(25)"B labirintus B"
1860 printtab(25)"[C= Z]CCCCCCCCCCCC
     [C= X]"
1870 print"[CRS LE][CRS LE][CRS LE]
     [CRS LE][CRS LE][CRS LE][CRS LE]
     [CRS LE]";
1880 printtab(25)"nyomj meg egy"
1890 printtab(106)"billentyut !"
1900 geta$:ifa$=""then1900
1910 return
1920 data169,0,141,57,192,141,63,192
1930 data141,191,2,169,4,141,58,192
1940 data169,216,141,64,192,162,0,189
1950 data251,197,208,1,96,141,188,2
1960 data232,189,251,197,141,189,2,232
1970 data189,251,197,141,190,2,173,188
1980 data2,201,255,208,3,76,65,192
1990 data141,60,6,173,189,2,141,60
2000 data218,206,190,2,240,62,56,173
2010 data57,192,105,39,144,6,238,58
2020 data192,238,64,192,141,57,192,141
2030 data63,192,173,58,192,201,7,208
2040 data32,56,173,57,192,201,111,144
2050 data24,169,4,141,58,192,169,216
2060 data141,64,192,169,0,109,191,2
2070 data141,57,192,141,63,192,238,191
2080 data2,76,46,192,173,24,192,56
2090 data105,2,144,21,141,24,192,141
2100 data34,192,141,41,192,238,25,192
2110 data238,35,192,238,42,192,76,21
2120 data192,141,24,192,141,34,192,141
2130 data41,192,76,21,192,20,0,15
2140 rem
2150 data32,0,1,223,6,2,160,6,21,105
2160 data6,2,32,0,2,223,6,2,160,6
2170 data19,105,6,2,32,0,4,223,6,2
2180 data160,6,17,105,6,2,32,0,6,223
2190 data6,2,160,6,15,105,6,2,32,0
2200 data8,223,6,2,160,6,13,105,6,2
2210 data32,0,10,223,6,2,160,6,11,105
2220 data6,2,32,0,12,223,6,2,160,6
2230 data9,105,6,2,32,0,14,223,6,2
2240 data160,6,7,105,6,2,32,0,16,223
2250 data6,2,160,6,5,105,6,2,32,0
2260 data18,223,6,2,160,6,3,105,6,2
2270 data32,0,20,223,6,2,160,6,1,105
2280 data6,2,32,0,21,233,6,2,160,6
2290 data1,95,6,2,32,0,20,233,6,2
2300 data160,6,3,95,6,2,32,0,18,233
2310 data6,2,160,6,5,95,6,2,32,0
2320 data16,233,6,2,160,6,7,95,6,2
2330 data32,0,14,233,6,2,160,6,9,95
2340 data6,2,32,0,12,233,6,2,160,6
2350 data11,95,6,2,32,0,10,233,6,2
2360 data160,6,13,95,6,2,32,0,8,233
2370 data6,2,160,6,15,95,6,2,32,0
2380 data6,233,6,2,160,6,17,95,6,2
2390 data32,0,4,233,6,2,160,6,19,95
2400 data6,2,32,0,2,233,6,2,160,6
2410 data21,95,6,1,0,32,0,2,160,1
2420 data21,32,0,1,0,255,0,255,255,0
2430 data209,32,0,2,160,1,21,32,0,1
2440 data0,255,0,139,32,0,3,160,1,7
2450 data32,1,3,255,0,14,32,0,2,160
2460 data1,7,32,1,2,0,255,0,255,255
2470 data0,62,32,0,2,160,1,7,32,1
2480 data2,255,0,14,32,0,3,160,1,7
2490 data32,1,3,0,255,0,185,160,1,6
2500 data255,0,18,160,1,6,255,0,18,160
2510 data1,6,255,0,18,160,1,6,255,0
2520 data18,160,1,6,255,0,18,160,1,6
2530 data0,160,1,255,160,1,230,0,32,1
2540 data255,32,1,230,0,255,0,47,32,0
2550 data4,160,1,13,32,0,11,160,1,13
2560 data32,0,11,160,1,13,32,0,2,0
2570 data255,0,255,255,0,125,32,0,2,160
2580 data1,13,32,0,11,160,1,13,32,0
2590 data11,160,1,13,32,0,3,0,255,0
2600 data185,32,0,2,160,1,5,32,0,2
2610 data0,255,0,255,255,0,41,32,0,2
2620 data160,1,5,32,0,2,0,255,0,207
2630 data32,0,2,160,1,5,32,0,19,160
```

```
2640 data1,5,32,0,19,160,1,5,32,0
2650 data19,160,1,5,32,0,2,0,255,0
2660 data116,160,1,13,32,0,11,160,1,13
2670 data32,0,11,160,1,13,32,0,11,160
2680 data1,13,32,0,11,160,1,13,32,0
2690 data11,160,1,13,32,0,11,160,1,13
2700 data32,0,11,160,1,13,32,0,11,160
2710 data1,13,32,0,11,160,1,13,32,0
2720 data11,160,1,13,32,0,11,160,1,12,0
5000 poke646,14:print"[SHF CL/HOME]"tab(
     12)"pontszamtablazat"
5005 printtab(11)"------------------"
5010 print"[CRS LE]":dimnev$(20):v=20480
5015 dimne(20):fori=1to9
5020 printi".>>>            <<< ..... >
     <"
5025 next
5030 print"10 .>>>            <<< .....
     >  <"
5035 print"[CRS LE]nev:>>> .......... <<
     < .....  >    <"
5040 fori=1topeek(827):fort=0to9
5045 e$=chr$(peek(v+t)):ne$(i)=ne$(i)+e$
     :nextt:v=v+10
5050 ne(i)=peek(v)*256+peek(v+1):v=v+3:
     nexti
5055 print"[CRS FEL][CRS FEL]":h=1:ne(11
     )=pt
5057 printtab(30)ne(11):print"[CRS FEL]
     [CRS FEL]"
5060 geta$:ifa$=""then5060
5062 ifa$=chr$(13)then5080
5063 ifa$=chr$(20)then5060
5065 ne$(11)=ne$(11)+a$:printtab(8)ne$(1
     1)
5070 print"[CRS FEL][CRS FEL]":h=h+1:ifh
     =11then5080
5075 goto5060
5080 ifne$(11)=""thenprint"[CRS LE]";:
     goto5055
5082 h=0:fori=1to10
5085 ifne(i)>=ne(i+1)thennext
5086 ifi=11then5095
5087 c=ne(i):c$=ne$(i):ne(i)=ne(i+1):ne$
     (i)=ne$(i+1)
5090 ne(i+1)=c:ne$(i+1)=c$:h=1:next
5095 ifh=1then5080
6000 print"[CL/HOME][CRS LE][CRS LE]
     [CRS LE]"
6005 fori=1to10:printtab(8)ne$(i),,ne(i)
6010 next
6020 v=20479
6025 fori=1to10:fort=1to10
6030 e$=mid$(ne$(i),t,1):ife$=""thene$="
     "
6032 pokev+t,asc(e$):nextt
6035 v=v+11:n1=int(ne(i)/256):n2=ne(i)-n
     1*256
6040 pokev,n1:pokev+1,n2:v=v+2:nexti
6050 print"[CRS LE][CRS LE]"
6055 printtab(6)"[CTRL 9] f [CTRL 0] -->
      a jatek folytatasa"
6060 print"[CRS LE][CRS LE]"tab(5)"disk
     parancsok a tablazatra :"
6065 print"[CRS LE]  [CTRL 9] s [CTRL 0]
      --> mentes          [CTRL 9] l
     [CTRL 0] --> toltes   "
6070 geta$:ifa$=""then6070
6075 ifa$="f"then6150
6080 ifa$="s"then6200
6090 ifa$="l"then6300
6100 goto6070
6150 clr:re=1:goto1
6200 poke16385,peek(45):poke16386,peek(4
     6):poke4*4096,1
6210 print"[SHF CL/HOME][CTRL 1][CRS LE]
     [CRS LE][CRS LE][CRS LE]poke43,0:po
     ke44,80:poke45,0:poke46,81[CRS LE]
     [CRS LE]"
6215 print"save"chr$(34)"@:pont
     lab"chr$(34)",8[CRS LE][CRS LE]
     [CRS LE]"
6220 print"[CRS LE][CRS LE][CRS LE]poke4
     3,1:poke44,8:poke45,peek(16385):pok
     e46,peek(16386):clr:gO0"
6225 print"[CL/HOME]"
6230 fori=0to7:poke631+i,13:next:poke198
     ,7:end
6300 clr:poke4*4096,1:print"[CRS FEL]
     [CRS FEL][CTRL 1]":load"pont
      lab",8,1
```

Vadnai Sz.:

PLUS/4-es programozói zsebkönyv

Az alapfogalmak tisztázása után precíz ismertetést kapunk a három beépített alkalmazói program lehetőségeiről. A programok teljes parancskészletét logikai sorrendben ismerteti. A könyv használhatóságát fokozza a végén lévő részletes tárgymutató, valamint a kapcsos, jól lapozható kötésforma.



Ára: 169,— Ft
Megjelent: 1987



Commodore PLUS/4 — A beépített programok kezelése

(Commodore)

A gép funkcióbillentyűivel négy különböző alkalmazói programot hívhatunk. A könyv megismerteti olvasóit e programok felhasználási lehetőségeivel, kezelésével. A megértést számos mintapélda bemutatása segíti.

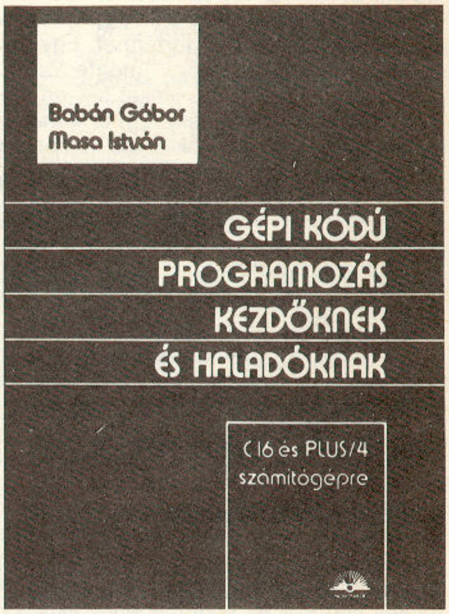


Babán G.—Masa I.:

Gépi kódú programozás kezdőknek és haladóknak

(C 16 és PLUS/4 számítógépre

A könyv első része tulajdonképpen tankönyv. Ebből a teljesen kezdők megismerkedhetnek az alapokkal.
A második rész alapos ismereteket nyújt a gépről. Részletesen bemutatják a szerzők az operációs rendszer lehetőségeit, valamint a BASIC interpreter működését. Külön fejezet foglalkozik a perifériák kezelésével.



Ára: 129,— Ft
Megjelent: 1987

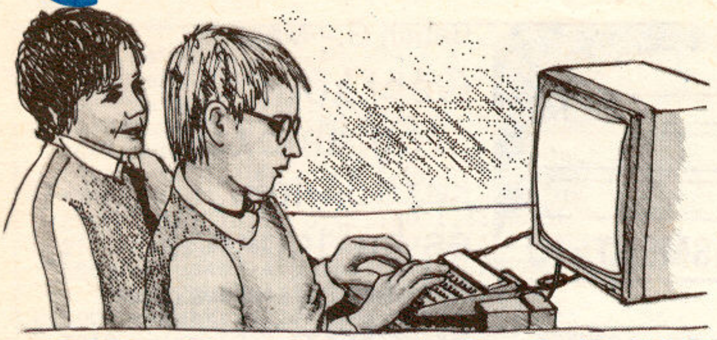


Commodore PLUS/4 — Felhasználói kézikönyv

(Commodore)

Ára: 99,— Ft
Megjelent: 1986

A PLUS/4-es gép tulajdonosainak nélkülözhetetlen segédkönyve. A legalapvetőbb ismeretek megszerzéséhez nyújt segítséget. Megtaláljuk benne a gép alapvető műszaki paramétereit is. A használhatóságot növeli az Ismerettár című fejezet, amelyben ábécérendben könnyen visszakereshetők a leglényegesebb tudnivalók.



PROGRAMOK

Vigyázat, vírus!

A számítógépes vírusok a számítástechnika világában időközben komoly fenyegetést jelentenek. Egy jóindulatú példány mutatja be, milyen fogásokat alkalmaznak ezek készítői.

Ha valaki vírusprogramot ír vagy nyilvánosságra hoz, tisztában kell lennie azzal, mi is történhet. Egy olyan vírus, amely bármilyen programokat képes „megfertőzni", képes arra is, hogy drága pénzen vett szoftvereket tegyen tönkre akkor is, ha nem is ez volt a készítő szándéka. Egy ártalmatlan vírusba kellemetlen kortársaink olyan rutinokat (pl. lemezformatálás) „olthatnak" be, amire az adott vírus mindjárt rosszindulatúvá válik. Mindezen szempontok alapján a „Mini Vírus" programunk (lásd a programlistát) elfogadható. A programlista egy „fertőzött" BASIC programot rejt. Az abban rejlő vírus képes BASIC programokat megfertőzni, és az a SAVE utasítással „szaporodik". Ezen kívül ezt a vírust könnyen el lehet távolítani, és a fertőzés ténye is könnyen megállapítható.

Egy BASIC programot nem is olyan könnyű vírusprogrammal megfertőzni. A számítógépünk tudvalevőleg mindig gépi kódban dolgozik, egy BASIC program nem más, mint egy halom adat, egy másik program számára. Ez a másik program a gép ROM-jában található BASIC interpreter. Ha megtekinthetnénk, hol tartózkodik a mikroprocesszor belső, úgynevezett programszámlálója a BASIC program futtatása alatt, akkor azt vagy a ROM-ban, vagy a nulláslap egy kis rutinjában találnánk. A ROM-ot azonban (szerencsére) nem lehet megfertőzni. A vírust azonban, amely maga is egy gépi kódú program, valahogy fel kell vinni. A mi esetünkben csakis a BASIC SYS utasítása jöhet szóba.

A SYS (a USR mellett) az egyetlen lehetőség arra, hogy elhagyjuk a BASIC interpretert. A BASIC program megfertőzésekor tehát be kell a programsorok közé csempészni egy SYS utasítást, amely azután felhívja a vírusunkat. Ezt a SYS-t azonban valóban végre kell hajtatni, azaz nem rakhatjuk el valami ritkán használt programba, vagy DATA sorok közé. Mivel szinte minden BASIC programot RUN-nal kell elindítani, megoldás a SYS utasítást a program legeslegelső utasításává tenni.

Természetesen nemcsak a SYS parancsot, de magát a vírusprogramot is el kell raknunk valahová. Annak nem szabad a változókat, sem egy esetleges gépi kódú segédrutint megzavarnia. A legegyszerűbb, ha azt is az első BASIC sorba írjuk. Hogy azzal az interpreter ne törődjön, egy REM utasítást kell eléje írni.

Természetesen a felhasználó igencsak csodálkozna, ha az általa írt programban egyszerre olyan sorokat találna, amelyeket nem írt meg. Éppen ezért a SYS utasítást és a vírust rejtő sorokat LIST ellen védeni kell. Ez egy második BASIC sorral és szintetikus vezérlő jelekkel történik.

A vírus tehát két BASIC sorban áll (0 és 1), amelyek védve vannak. A tudatlan felhasználó sem a LISTázásnál, sem a program futása közben nem veszi észre azt. Mihelyt a programot egy NEW-val töröltük, vagy egy másik programot behívunk, a vírus maga is eltűnik a BASIC programmal együtt. A szaporodás veszélye ezzel jelentősen lecsökken.

A vírus első akciója az indítás után, hogy a BASIC terület kezdete címet magasabb értékre állítja, méghozzá pontosan a harmadik BASIC sor elejére. Ez maga az „eredeti" program első sora, így tehát minden rendben futhat tovább. Most azonban a vírus védve lesz NEW és LOAD ellen. Most azonban a vírus már nem tartozik a programhoz, így azt SAVE esetén nem mentenénk ki adathordozóra. Enélkül pedig nincs szaporodás.

Éppen ezért a vírus egy SAVE utasításnál visszaállítja a vektorokat az eredeti értékekre. Az ismételt betöltéskor tehát ott lesz a vírus a BASIC programunk első két sorában. Ez akkor is így lesz, ha közben egy másik programot adtunk vagy hívtunk be.

A C64-es tárolójában van egy vektor, amelyre a gép minden SAVE esetében hivatkozik. A vírus második akciója, hogy ezt a vektort önmaga egy részére irányítja. A vírusnak ezt a rutinját tehát minden SAVE esetében végrehajtatjuk. Itt állítjuk vissza az eredeti BASIC start címeket, majd a normál SAVE rutinra ugrunk vissza. Így a vírus az adott programmal együtt adathordozóra kerül.

Ebben a rutinban természetesen előbb ellenőrizni kell, hogy a tárolóban álló program nem fertőzött-e már. Ebben az esetben

```
1 rem  ********************************
2 rem" * C= Ujsag           Ssz.:164    *
3 rem" *           MINI VIRUS           *
4 rem" * Program:     A 64'er nyoman    *
5 rem  ********************************
6 rem" * Betoltes elott:   POKE 43,1    *
7 rem" *   POKE 44,12: POKE 12*256,0    *
8 rem" * Futtatas utan:    POKE 43,1    *
9 rem" *  POKE 44,8: POKE 45,227        *
10 rem"*  POKE 46,10  & SAVE'MVIRUS',8*
11 rem ********************************
12 print chr$(147)"data beolvasas es ellenorzes ...";:j=02049:ve=02786:p=j
13 for b=0 to 7:read a$
14 l=asc(mid$(a$,2,1))
15 h=asc(mid$(a$,1,1))
16 l=l-48:if l>9 then l=l-7
17 h=h-48:if h>9 then h=h-7
18 print"[illegible]"p;:p=p+1
19 if h>15 or l>15 then 21
20 a=h*16+l:poke j+b,a:t=t+a:next b:read a:if a=t then 22
21 print:print"data hiba ...  sor:"peek(64)*256+peek(63):end
22 t=0:j=j+8:if j<ve then 13
23 print"[illegible]kesz":end
24 data bd,08,00,00,00,3a,3a,3a,0371
```

```
25 data 3a,9e,32,30,36,35,3a,8f,0622
26 data a9,c9,85,2b,a9,08,85,2c,0900
27 data a9,30,8d,32,03,a9,08,8d,0729
28 data 33,03,a2,01,a9,81,9d,ff,0927
29 data 02,a9,08,8d,01,03,60,ad,0593
30 data c9,08,c9,85,f0,47,ad,cd,1232
31 data 08,c9,9e,f0,40,a5,b9,29,1062
32 data 0f,d0,3a,a9,01,85,c1,a9,0946
33 data 08,85,c2,a0,04,b9,af,08,0867
34 data c9,20,d0,02,a9,30,aa,e8,1062
35 data e0,3a,d0,0a,a9,30,99,af,1045
36 data 08,88,10,e9,c8,aa,8a,99,1054
37 data af,08,ce,a2,08,d0,0e,ad,0954
38 data a3,08,8d,a2,08,ee,13,03,0742
39 data a9,07,8d,a1,08,4c,ed,f5,1044
40 data 8a,48,20,11,08,ad,13,03,0462
41 data f0,11,ce,a1,08,d0,0c,a9,1021
42 data 02,8d,a1,08,a9,a4,a0,08,0813
43 data 20,1e,ab,68,aa,4c,8b,e3,0949
44 data 03,03,03,0d,4d,49,4e,49,0323
45 data 56,49,52,55,53,3a,20,20,0531
46 data 20,20,31,20,43,4f,50,49,0444
47 data 45,53,21,00,c9,08,01,00,0395
48 data 8f,22,14,8d,91,91,91,00,0773
49 data cf,08,64,00,8f,00,d5,08,0679
50 data 65,00,8f,00,f4,08,66,00,0598
51 data 8f,20,20,20,54,4f,52,5a,0574
52 data 53,54,45,4e,59,45,5a,4f,0641
53 data 4b,52,45,20,42,4f,4e,54,0565
54 data 41,53,00,fa,08,67,00,8f,0652
55 data 00,00,09,68,00,8f,00,0a,0266
56 data 09,6e,00,99,22,93,22,3b,0546
57 data 00,28,09,78,00,99,3a,99,0533
58 data 22,20,41,20,46,45,4c,42,0444
59 data 4f,4e,54,41,4e,44,4f,20,0563
60 data 53,5a,41,4d,22,3b,00,30,0456
61 data 09,82,00,85,20,5a,00,49,0467
62 data 09,8c,00,99,22,20,54,4f,0531
63 data 52,5a,53,54,45,4e,59,45,0644
64 data 5a,4f,4b,3a,20,22,3b,00,0427
65 data 51,09,96,00,51,b2;5a,00,0589
66 data 69,09,a0,00,54,b2,32,3a,0644
67 data 50,b2,30,3a,8d,20,35,30,0638
68 data 30,3a,8d,20,36,30,30,00,0429
69 data 81,09,aa,00,54,b2,33,3a,0679
70 data 50,b2,30,3a,8d,20,35,30,0638
71 data 30,3a,8d,20,36,30,30,00,0429
72 data 98,09,b4,00,8b,20,54,b1,0773
73 data ba,28,51,29,aa;31,20,a7,0766
74 data 20,89,20,32,30,30,00,b8,0531
75 data 09,be,00,54,b2,54,aa,32,0765
76 data 3a,50,b2,30,3a,8d,20,35,0648
77 data 30,30,3a,8d,20,36,30,30,0477
78 data 3a,89,20,31,38,30,00,d2,0590
79 data 09,c8,00,54,b2,51,3a,50,0690
80 data b2,31,3a,8b,20,51,b1,31,0763
81 data 20,a7,20,8d,20,36,30,30,0554
82 data 00,ef,09,d2,00,8b,20,51,0710
83 data b2,5a,20,a7,20,99,a3,32,0865
84 data 31,29,3b,22,50,52,49,4d,0495
85 data 53,5a,41,4d,22,00,f9,09,0607
86 data dc,00,89,20,31,32,30,00,0536
87 data ff,09,f4,01,8f,00,05,0a,0667
88 data f5,01,8f,00,22,0a,f6,01,0680
89 data 8f,20,20,20,54,45,4e,59,0559
90 data 45,5a,4f,4b,20,45,4c,4c,0566
91 data 45,4e,4f,52,5a,45,53,45,0619
92 data 00,28,0a,f7,01,8f,00,2e,0487
93 data 0a,f8,01,8f,00,54,0a,fe,0750
94 data 01,8b,20,54,ac,b5,28,51,0730
95 data ad,54,29,b2,51,20,a7,3a,0814
96 data 20,51,b2,51,ad,54,3a,50,0767
97 data b2,50,aa,31,3a,89,20,35,0757
98 data 31,30,00,5a,0a,08,02,8e,0349
99 data 00,60,0a,58,02,8f,00,66,0441
100 data 0a,59,02,8f,00,84,0a,5a,0476
101 data 02,8f,20,20,20,54,4f,52,0486
102 data 5a,53,54,45,4e,59,45,5a,0652
103 data 4f,4b,20,4b,49,49,52,41,0554
104 data 53,41,00,8a,0a,5b,02,8f,0532
105 data 00,90,0a,5c,02,8f,00,a2,0553
106 data 0a,62,02,8b,20,50,b2,30,0587
107 data 20,a7,20,89,20,36,34,30,0554
108 data 00,d5,0a,6c,02,99,a3,32,0699
109 data 31,29,3b,c9,28,22,20,20,0488
110 data 20,20,20,20,20,20,20,20,0256
111 data 22,aa,c4,28,54,29,2c,38,0665
112 data 29,3b,3a,8b,20,50,b1,31,0635
113 data 20,a7,20,99,22,20,5e,22,0578
114 data 3b,50,3b,00,db,0a,76,02,0547
115 data 99,00,e1,0a,80,02,8e,00,0660
116 data 00,00,00,00,00,00,00,00,0000

ready.
```

nem szabad újrafertőzni. A programhoz ekkor nem nyúlunk. Ezenkívül azt is ellenőrizni kell, hogy BASIC programról van-e szó. A vírus ezt akkor veszi adottnak, ha a tároláskor másodlagos cím megadása nélkül (SAVE "név",8) mentjük ki a programot. Ha másodlagos címmel SAVE-elünk (SAVE "név",8,1), akkor szintén nem nyúlunk a programhoz.

A Mini Vírus egy harmadik vizsgálatot is végez. Nagy számban vannak ugyanis olyan gépi kódú programok, amelyeket normál BASIC programként kell betölteni, s amelyek SYS utasítással kezdődnek. Az ilyen programokat a vírus nagy valószínűséggel tönkretenné, mert a sorbeszúrások miatt a címek nem stimmelnek többet. Hogy kárt ne okozzunk, ezért azokat a programokat sem fertőzzük meg, amelyek egy SYS utasítással kezdődnek. Ez mindjárt védekezési módot is kínál a vírussal szemben. Ha ugyanis a BASIC program első utasítása egy „ártalmatlan" SYS parancs, fertőzés nem történik. Ilyen utasítás lehet például a SYS 138 (Assembler programozók tudhatják, ezen a címen áll a CHR-GET rutin utolsó utasítása, egy RTS).

A SAVE vektort azonban a ‹RUNT/STOP› ‹RESTORE› visszaállítja a normál értékekre. Ezzel pedig a vírust hatástalaníthatnánk.

Vannak azonban más vektorok is. Ezek közül egyet minden egyes READY jelzés kiadásakor használunk. Ez az úgynevezett melegindítás vektor. Ezt a ‹RUN/STOP› ‹RESTORE› billentyűk megnyomásával sem tudjuk elállítani. A vírus harmadik akciója tehát, hogy ezt a vektort is saját magára irányítja. Ebben a rutinban pedig arra „vigyázunk", hogy a SAVE vektor mindig a megfelelő vírusrutinra mutasson. Ezenkívül a vírus bizonyos feltételek mellett bejelentkezik, mielőtt a régi melegindításra ugrana. A vírus tehát az alábbi részekből áll:

— Keret két BASIC sorból.
— Inicializálás (SYS-szel):
1. BASIC startot eltolni.
2. SAVE vektort átállítani.
3. A melegindítás vektort átállítani.
— SAVE rutin (minden egyes SAVE esetében):
1. Ellenőrizni, hogy fertőzött-e már a program (ha igen, normál SAVE).
2. Ellenőrizni, hogy az első utasítás a SYS-e (ha igen, normál SAVE).
3. Ellenőrizni, hogy a program BASIC program-e (ha igen, normál SAVE).
4. A BASIC tárolót visszaállítani az eredeti értékekre.
5. A másolatok számát jelző értéket eggyel megnövelni.
6. Minden harmadik másolatnál kijelzés.
7. A SAVE rutinra ugrani.
— melegindítás rutinra (minden READY-nél):
1. Az inicializálást megismételni.
2. A vírust kijelezni, ha ez megengedett.
3. A melegindításhoz ugrani.

A vírust egy reset (pl. SYS 64738) „megöli". A fertőzött programok esetében pedig elegendő a 0 és az 1-es sorokat törölni.

Ha szeretnénk ellenőrizni, hogy a vírus aktív-e a számítógépben, elég egy egyszerű POKE utasítást: POKE 787,9-et adni. Ettől a pillanattól kezdve (az aktív vírus esetében) minden második READY után kiadjuk az eddig elkészített vírusmásolatok (azaz a fertőzött programok) számát.

Nagyon vigyázzunk mindenféle manipulációval. Egy ilyen vírus fegyver, még akkor is, ha nem írnak bele lemezformatáló rutint! Egy vírussal mindig a legnagyobb felelősséggel kell bánni. Felelőtlen dolog (és jogellenes is) például a vírust úgy továbbadni, hogy a címzettet erről nem informáljuk.

HÍREK

Új és olcsóbb Voice Master

A C64-esen eleddig csak a Voice Masterrel lehetett igazán jó minőségben hangdigitalizálást megvalósítani. Most ezt a csodálatra méltó bővítőt kompakt formában is megvásárolhatjuk. Az jelenleg csupán egy kézbeillő házból, mikrofonból és egy, a felvételi hangerőszabályozóból áll.

A Voice Master lehetővé teszi beszéd felismerését és digitalizálását a C64-gyel. Azaz adott a beszédvezérlésű programok készítésének lehetősége is. A csomagban mellékelt Blackjack játék mindjárt példát is mutat erre. Mielőtt ezzel a programmal elkezdhetnénk játszani, a játékosnak mikrofonba kell mondani a szükséges szavakat. A C64-es letárolja a hangot. Ezután a játékosnak csak a kívánt parancsot (pl. lapot adni, játék, stb.) kell kimondania. A C64-es összeveti a hangbevitelt a tárolt információkkal, és végrehajtja amit kérünk.

A Voice Master már évek óta népszerű a C64-es tulajdonosok között. Eleddig azonban erősen visszatartó volt a készülék ára. Az új Voice Master Junior nemcsak méretben, hanem árban is zsugorodott. Most 100 (korábban 250!) márkáért már birtokba vehetjük a kiegészítőt. A jellemzőkön szerencsére nem spóroltak a tervezők.

Info.: Print-Technik, Nikolaistrasse 2, D—8000 München 40.

VESZTESÉGES A NORSK DATA

A skandináv számítástechnikai ipar egyik vezető cégének tartott Norsk Data váratlanul veszteséget jelentett a legutóbbi mérlegbeszámolóján. A hír önmagában nem érdemelne sok figyelmet, a hátterében meghúzódó körülmények miatt azonban több szempontból is tanulságos esetről van szó. A közepes méretű vállalat nehézségei elsősorban azt példázzák, hogy a nagy horderejű piaci változások hatására miként sorvadnak el rövid idő alatt a legdinamikusabbnak tartott ágazatban is a stabil pozíciók.

A Norsk Data kedvezőtlen eredményei két fontosabb eredőre vezethetők vissza. Ezek közül az egyik lényegében gazdaságföldrajzi és piaci feltételekből fakad, a másik pedig a műszaki fejlesztési irányok helytelen megválasztásából. Az ipari szerkezetváltási törekvések szempontjából mindkettő komoly tanulságokkal szolgál.

A regionális, piaci jellegű feltételek elemzése kapcsán a szakemberek elsősorban arra mutatnak rá, hogy a Norsk Data egy olyan térségben tevékenykedett, ahol a jól képzett szakembergárda, a magas színvonalú ipari kultúra és a fejlett infrastruktúra egyaránt rendelkezésre állt. A vállalat elsődlegesen egy olyan piacon volt jelen, amely folyamatosan magas követelményszintet támasztott, de jól fizetett a színvonalas termékekért és szolgáltatásokért, így az árbevétel bőséges fedezetet nyújtott az intenzív műszaki fejlesztésre.

A kedvező gazdasági környezet és a vélhetően igényes vállalatvezetés sem volt azonban elégséges a folyamatos expanzió fenntartásához. A skandináv térség, illetve az eladási lehetőségek szempontjából szóbajöhető nyugat-európai országok csak viszonylag szűk regionális piacnak minősültek, ahonnan nem lehetett versenybe szállni a számítástechnika fejlődésének fő irányait meghatározó amerikai és japán nagyvállalatokkal. A piac nyitottságára építő állami szabályozás is a Norsk Data ellen dolgozott egy idő után. Amint ugyanis kiderült, hogy az eredeti műszaki fejlesztési irány zsákutcába jutott, a vállalat védtelen maradt a konkurrenciával szemben, és eladásai rohamosan csökkenni kezdtek.

A vállalat helyzetének rosszabbra fordulása ezen a ponton kapcsolódik a műszaki fejlesztési irány helytelen megválasztásához. A Norsk Data úgynevezett miniszámítógépek gyártására szakosodott az évtized kezdete óta. (A szakterületet kevésbé ismerő olvasók számára ezt a kategóriát elsősorban a kisebb teljesítményű asztali személyi számítógépektől szokták megkülönböztetni. A miniszámítógépek a folyamatos miniatűrizálás és méretcsökkentés ellenére még ma is többnyire állványos kivitelben készülnek, és akkorák, mint egy kisebb irattároló szekrény.)

Korábban a Norsk Data egy olyan fejlesztési irányt szabott meg, amely éveken keresztül ragyogó üzletmenetet biztosított számára. A cég teljesen elszakította a konstrukciót azoktól az alapmodellektől, amelyeket a nagy amerikai gyárak követtek. Az önálló út követése elsősorban abban nyilvánult meg, hogy a cég gépeit teljesen egyedi alapszoftverrel (úgynevezett operációs rendszerrel) látták el. Értelemszerűen következett ebből az, hogy a felhasználói programok csak ezeken a gépeken működtek.

Ez a fejlesztési stratégia jól működött a 80-as évek elejéig, amikor a piac még honorálta a drágább egyedi megoldásokat. A Norsk Data ebben az időben káprázatos nyereséget realizált az általa forgalmazott programcsomagokban és egyedi szoftvertermékeken. Ez a stratégia azonban az évtized végének közeledtével elkerülhetetlenül bukásra lett ítélve. A fordulatot az váltotta ki, hogy az utóbbi években vdratlanul előretörtek az úgynevezett nyilt számítógépes architektúrák, amelyekben számos ponton nemzetközileg szabványosított megoldásokat alkalmaznak.

Az általános, szabványosított kellékek sorában a legfontosabbnak azt a UNIX nevű operációs rendszert tartják, amelyet még a 70-es évek végén fejlesztett ki az AT & T Bell Laboratóriuma. Ez az alapszoftver, amelyet egészen a mai napig folyamatosan tökéletesítenek, elsősorban arra nyújt jó lehetőséget, hogy az alkalmazói programokból kialakuljon egy olyan, nemzetközi méretű, közös sz9ftverkincs, amelyből a felhasználók bőségesen meríthetnek. Magától értetődik, hogy minden olyan cég, amely a saját partikuláris megoldásait ajánlja egy ilyen irányzattal szemben egy idő után elkerülhetetlenül zsákutcába jut.

A Norsk Data eredményeinek váratlan és rendkívül erőteljes romlása mögött is éppen az áll, hogy nem ismerte fel kellő időben a nagy nemzetközi szabványokhoz való igazodás szükségességét. Más tekintetben a vállalat sorsa kezd emlékeztetni arra a pályára, amelyet a Saab-Scania számítástechnikai részlege futott be. Ezt ugyanis a sorozatos balsikerek után átvette az elektronika más területein sikeres Ericsson, majd egy további eredménytelen periódus, sőt tetemes veszteségek után továbbadta a finn Nokiának.

E példák egybehangzóan azt sugallják, hogy a legmodernebb ágazatokban való megkapaszkodás kulcskérdése ma az, megtalálja-e egy vállalat a helyét a mélyreható szabványosítások által meghatározott nemzetközi piaci folyamatokban. (AP—DJ).

Tippek trükkök a C128-hoz



VDC WINDOW

Ez a rutin lehetővé teszi, hogy a 80 karakteres képernyőt bemásoljuk a VDC chip szabad tárolójába, majd azt onnan visszahozzuk. Itt még az attribut tárolót is figyelembe vesszük. Adjuk be a listát 64-es módban. A példaprogramban előbb egy képet készítünk, majd azt a tárolóba mentjük. Most az eredeti képet új utasításokkal meg lehet változtatni. Ezután nem kell a régit elkészíteni, a képet egy szemvillanás alatt visszaállíthatjuk. Beugrási címek:
80 karakteres kép SAVE: $1303
80 karakteres kép LOAD: $1300

```
vdc-window - c128

1 rem **********************************
2 rem *                                *
3 rem *       vdc-window c128    165   *
4 rem *                                *
5 rem **********************************
6 :
7 :
8 print chr$(147)"data beolvasas es ellenorzes ...";:j=04864:ve=05007:p=j
9 for b=0 to 7:read a$
10 l=asc(mid$(a$,2,1))
11 h=asc(mid$(a$,1,1))
12 l=l-48:if l>9 then l=l-7
13 h=h-48:if h>9 then h=h-7
14 print"[illegible]"p;:p=p+1
15 if h>15 or l>15 then 17
16 a=h*16+l:poke j+b,a:t=t+a:next b:read a:if a=t then 18
17 print:print"data hiba ...  sor:"peek(64)*256+peek(63):end
18 t=0:j=j+8:if j<ve then 9
19 print"[illegible] kesz":end
20 data 4c,55,13,a2,18,8e,00,d6,0722
21 data 2c,00,d6,10,fb,ad,01,d6,0913
22 data 09,80,48,20,79,13,a0,10,0557
23 data a9,00,85,fa,85,fb,85,fd,1322
24 data a9,10,85,fc,a5,fc,a2,12,1167
25 data 20,79,13,a9,00,e8,20,79,0726
26 data 13,a5,fa,a2,20,20,79,13,0800
27 data a9,00,e8,20,79,13,a2,1e,0765
28 data a9,00,20,79,13,e6,fa,e6,1051
29 data fc,88,d0,d8,68,29,7f,a2,1246
30 data 18,20,79,13,60,a2,18,8e,0620
31 data 00,d6,2c,00,d6,10,fb,ad,0912
32 data 01,d6,09,80,48,20,79,13,0596
33 data a0,10,a9,00,85,fc,85,fd,1116
34 data 85,fb,a9,10,85,fa,4c,24,1064
35 data 13,8e,00,d6,2c,00,d6,10,0649
36 data fb,8d,01,d6,60,00,00,00,0703
37 data 00,00,00,00,00,00,00,00,0000

ready.
```

```
vdc-window-demo

1 bank15
2 fast
3 graphic5,1
4 printchr$(19)chr$(19)chr$(27)"m";
5 :
6 fori=1to2000
7 : ifrnd(0)>.5thenprintchr$(15);:elseprintchr$(143);
8 :  color5,int(rnd(1)*15)+1
9 : printchr$(205+rnd(0)+.5);
10 next
11 :
12 print"chr$(19)chr$(19)
13 :
14 sys dec("1303"):rem kep tarolasa
15 :
16 printchr$(27)"l"chr$(5)chr$(143);
17 :
18 :
19 char1,30,12,"*********************"
20 char1,30,13,"*                   *"
21 char1,30,14,"*     billentyu     *"
22 char1,30,15,"*                   *"
23 char1,30,16,"*   a regi kephez   *"
24 char1,30,17,"*                   *"
25 char1,30,18,"*                   *"
26 char1,30,19,"*********************"
27 getkeya$
28 :
29 sys dec("1300"):rem a kep ujra eloallitasa
30 :
31 sleep1
32 goto19

ready.
```

Címkeíró C(64)-re

Bizonyára sokaknak okozott gondot a PROFI-ASS 64 használata közben az, hogy nem tudja megnézni a szimbólum-táblázatot. Erre ad megoldást a következő gépi kódú rutin, melynek kezdőcíme $C900 (51456). (Természetesen máshova is lehet fordítani).

A rutin használata:

— töltsünk a gépbe egy „PROFI-ASS 64"-et (,8,1)
— SYS 64738 (teljes reset),
— írjuk be (ha megvan, töltsük be) a címkekiíró programot,
— RUN (a program lefordul az 51456-os címre),
— forrásprogram betöltése és fordítása,
— SYS 51456 (a rutin hívása).

Ekkor kiíródnak a címkék és értékeik, de fordított sorrendben. Ez azért van, mert a PROFI-ASS a címkéket BASIC stringként kezeli, ezért az egyértelmű kiíratás csak ebben a sorrendben lehetséges.

Vigyázni kell arra, hogy az 51456-os címre ne fordítsunk semmit, mert „elgázolná" ezt a kis rutint.

```
1 REM *********************************
2 REM *    C= UJSAG SORSZAM:166        *
3 REM *    CIMKEKIIRO C64              *
4 REM *    PROGRAM: MOLNAR LASZLO      *
5 REM *********************************
100 SYS36864
110 .OPT OO,P
120 *=$C900                 ;ML 1050
130 :         LDA #00
140 :         STA $F9
150 :         LDX $33       ;$FA-$FB CIMEK
160 :         LDA $34       ;  KEZDOCIME
170 :         STX $FA
180 :         STA $FB
190 :LP0      CMP #$90      ; $9000 PRINT
200 :         BEQ VEG
210 :         LDY #01
220 :LP1      LDA ($FA),Y
230 :         STA $FC,Y
240 :         DEY
250 :         BPL LP1
260 :         LDA $F9
270 :         AND #01
280 :         BEQ LP2
290 :         LDY #20       ; KIIRAS A 20.
300 :          .BYT 12
310 :LP2      LDY #00       ; KIIRAS A 0.
320 :         STY 211       ; POZICIOTOL
330 :         LDY #02
340 :         LDA ($FA),Y
350 :         TAX
360 :         AND #07
370 :         STA $FE
380 :         TXA           ; CIMKE HOSSZANAK
390 :         CLC           ; KISZAMITASA
400 :         ADC #08
```

```
410 :        AND #$F8
420 :        LSR
430 :        LSR
440 :        LSR
450 :        ADC #$40
460 :        JSR $FFD2
470 :        LDA $FE
480 :        BEQ LP4
490 :        TAX
500 :LP3     INY
510 :        LDA ($FA),Y
520 :        JSR $FFD2
525 :        DEX
530 :        BNE LP3
540 :LP4     LDA 211
550 :        CLC
560 :        ADC #08
570 :        SEC
580 :        SBC $FE
590 :        STA 211
600 :        LDA #"="   ; = KIIRASA
610 :        JSR $FFD2
620 :        LDA $FD    ; ERTEK KIIRASA
630 :        JSR SZK0
640 :        LDA $FC
650 :        JSR SZK0
660 :        INC $F9
670 :        LDA $F9
680 :        LSR
690 :        BCS LP5
700 :        LDA #13    ; RETURN
710 :        JSR $FFD2
720 :LP5     INC $FE
730 :        INC $FE
740 :        LDA $FA    ; KOVETKEZO CIMKE
750 :        SEC        ; CIMENEK
760 :        ADC $FE    ; KISZAMOLASA
770 :        STA $FA
780 :        BCC LP6
790 :        INC $FB
800 :LP6     JSR $F6ED ; RUN-STOP
810 :        BEQ VEG
820 :        LDA $FB
830 :        JMP LP0
840 :VEG     RTS
850 :SZK0    TAX        ; 1 BYTE-OS SZAM
860 :        LSR        ; KIIRASA
870 :        LSR
880 :        LSR
890 :        LSR
900 :        JSR SZK1
910 :        TAX
920 :SZK1    AND #$0F
930 :        CMP #$0A
940 :        BCC SZK2
950 :        ADC #06
960 :SZK2    ADC #$30
970 :        JMP $FFD2
980 .END

READY.
```

Piros-fehér-zöld háttér

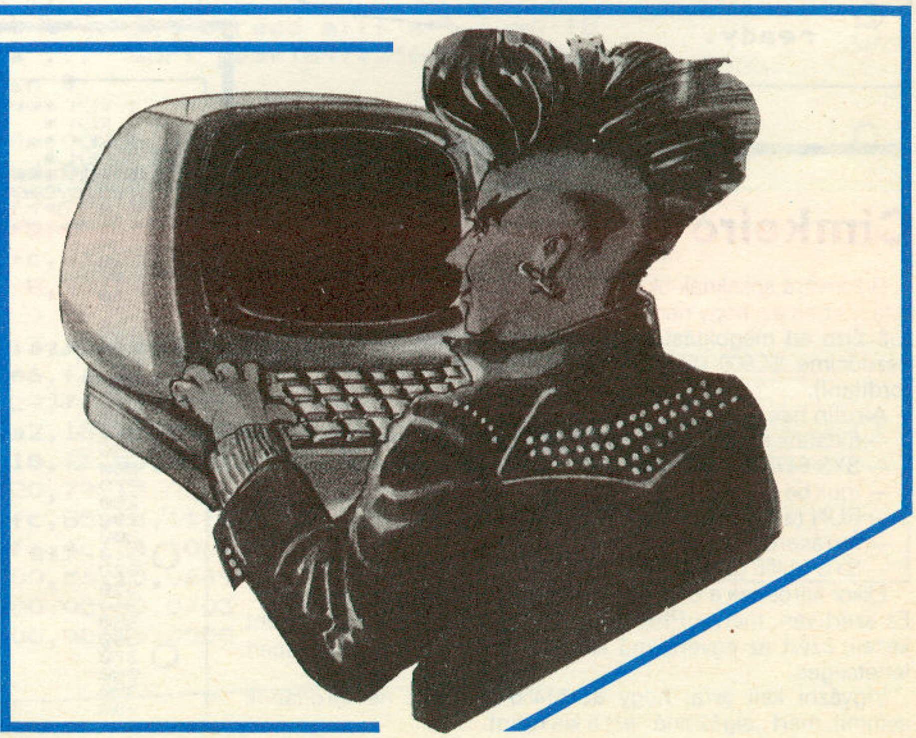
A program a videovezérlő megszakítási lehetőségeit felhasználva a képernyőre varázsolja a magyar nemzeti színeket úgy, hogy azok a későbbi munkát nem zavarják, és csak a STOP-RESTORE megnyomásával vagy a rasztermegszakítás kikapcsolásával lehet őket eltűntetni.

A 3. lista begépelése és a PROFI-ASS 64-gyel való lefordítása után a SYS 49152 utasítással lehet bekapcsolni a piros-fehér-zöld képernyőt.

```
1 REM ***********************************
3 REM *  PIROS-FEHER-ZOLD KEPERNYO C64 *
4 REM *  PROGRAM: SOMOS PETER, 1988.   *
5 REM ***********************************
10 SYS (36864)                    167
15 ;
30 ;
45 ;
49 .OPT P,OO
50 *= $C000
60 IRQVECT = $0314 ; MEGSZAKITAS-VEKTOR
70 VIC = $D000     ; VIDEOVEZERLO KEZDOCIME
80 IRQRET = $EA31  ; MEGSZAKITAS KEZDOCIME
90 CIA1 = $DC00    ; CIA1 KEZDOCIME
100 RETIRR = $FEBC
110 RED = 20
120 WHITE = 106
130 GREEN = 197
200 ; +-+-+-+-+-+-+
210 SEI              ; INICIALIZALAS
220 LDA #<Q1:STA IRQVECT
230 LDA #>Q1:STA IRQVECT+1
240 LDA #RED:STA VIC+18
250 LDA VIC+17:AND #$7F:STA VIC+17
260 LDA #%10000001:STA VIC+26
270 CLI
280 RTS
290 ; +-+-+-+-+-+-+
300 Q1 LDA VIC+25. ; UJ IRQ-RUTIN
301 STA VIC+25:BMI Q2
310 LDA CIA1+13
320 CLI
330 JMP IRQRET
335 ; +-+-+-+-+-+-+
340 Q2 LDA VIC+18:CMP #GREEN:BCS Q4
350 CMP #WHITE:BCS Q3
355 ; +-+-+-+-+-+-+
360 LDA #2:STA VIC+32:STA VIC+33
370 LDA #WHITE:STA VIC+18
380 JMP RETIRR
385 ; +-+-+-+-+-+-+
390 Q3 LDA #1:STA VIC+32:STA VIC+33
400 LDA #GREEN:STA VIC+18
410 JMP RETIRR
415 ; +-+-+-+-+-+-+
420 Q4 LDA #5:STA VIC+32:STA VIC+33
430 LDA #RED:STA VIC+18
440 JMP RETIRR

READY.
```

HÍREK



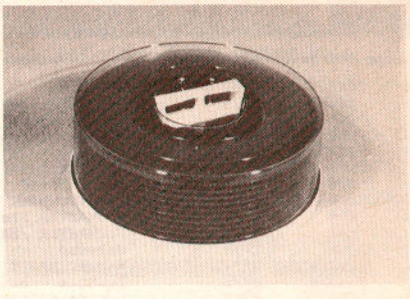
Új MIDI interfész

A Rössmöller Handshake GmbH most egy C64-es MIDI interfészt is kínál. Ez egy bemenettel (MIDI-in), egy átvezetett (MIDI-Thru), és négy szabályos (MIDI-Out) kimenettel rendelkezik. Ezzel tehát maximum hat szintetizátor párhuzamos csatlakoztatása lehetséges a C64-hez. A gyártó adatai szerint az interfész kompatibilis a szabvány szoftverekhez, pl. a C-Lab, a Steinberg és a Jellinghaus termékekhez. Az ára: 99 márka.

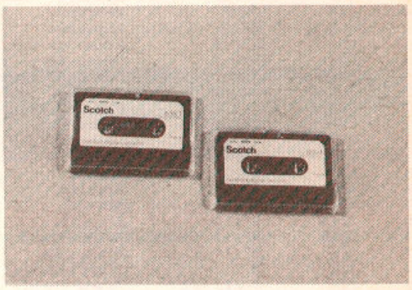
Info.: Rosmöller Handshake GmbH, Neuer Markt 21, D—5309 Meckenheim.

ÁPISZ számítástechnikai szaküzlet
VIII. Szigony u. 15. 143-446

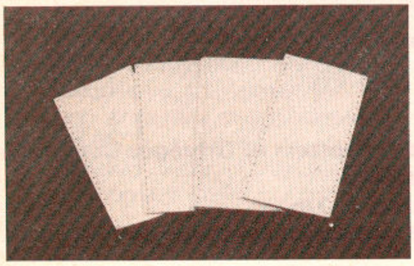




60 Mbyte-os mágneslemez **24 900 Ft**



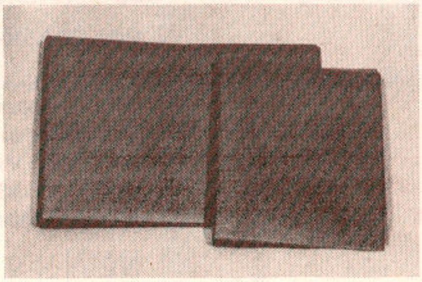
3M digitál kazetta **690 Ft**

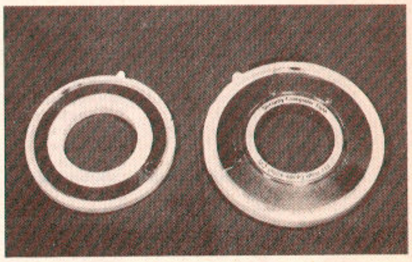


Leporellók: — egy és több példányos
— kis és nagy nyomtatókhoz

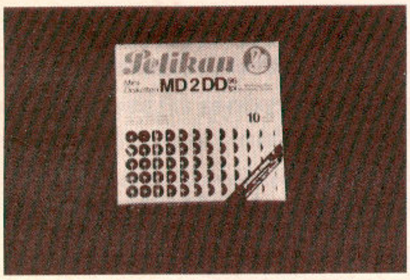
Etikett leporellók
120×48 mm-es egypályás, 1 doboz
3000 címke — **2000 Ft**

8 collos floppytároló PVC mappa **280 Ft**





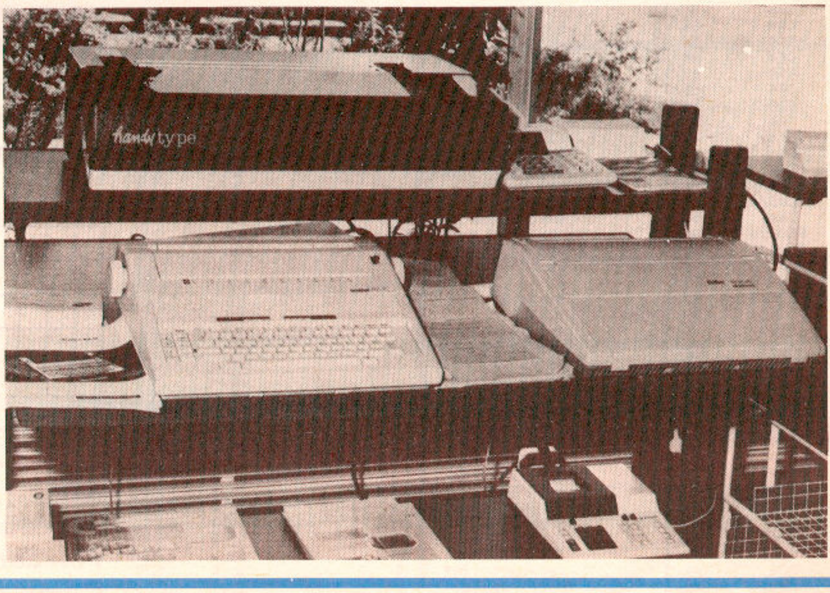
Mágnesszalagok 600—3600 láb hosszú
700 Ft-tól 3700 Ft-ig



5¼ collos floppy lemezek
IBM XT és AT gépekhez
150 Ft-tól 590 Ft-ig

SZÁMÍTÁSTECHNIKA
Ápisz
SZÁMÍTÁSTECHNIKA

Brother AX—15 elektromos táskaírógép	34 500 Ft
Erika 3004 elektromos táskaírógép	22 700 Ft
Handy type sornyomtató	15 000 Ft
Sharp 2607S asztali számológép	11 800 Ft
ELWRO asztali számológép	2 900 Ft



PÖTYÖGŐ SZOLGÁLAT

sorszám	géptipus	megjelenés	elnevezés	ára
050	C16	87/2	Telezsák	60,-
051	C64	87/3	Stringkeresö	20,-
052	C16	87/3	Jövedelemadó	40,-
053	C16	87/3	Hangos billentyük	20,-
054	VC20	87/3	Rajzoló	20,-
055	PLUS/4	87/4	Karaktervezö	50,-
056	C64	87/4	Dupla BASIC	20,-
057	VC20	87/4	Ferde torony	40,-
058	C64	87/5	GEOMASTER /leforditott/	60,-
059	C64	87/5	GEOMASTER /BASIC változat/	50,-
060	C64	87/5	Golyózó	50,-
061	PLUS/4	87/5	Magyar BASIC	50,-
062	PLUS/4	87/5	Kukkoló - BASIC monitor	30,-
063	C64	87/6	Sprite editor +	50,-
064	C64	87/6	Magnó directory	60,-
065	C16	87/6	Nagyfelbontású hardcopy	50,-
066	C16	87/6	Sztereó vizió	20,-
067	C16	87/6	Kukkoló - Szuper peek	40,-
068	C16	87/7	Mentööv	20,-
069	PLUS/4	87/7	Ablakozó	50,-
070	C64	87/8-9	Commo-do-re /BASIC változat/	60,-
071	C64	87/8-9	Commo-do-re /leforditott/	60,-
072	PLUS/4	87/8-9	Kukkoló - Bövitett monitor	30,-
073	PLUS/4	87/8-9	Ékezetes betük	30,-
074	PLUS/4	87/8-9	PLUTO - Rajzoló program	60,-
075	VC20	87/8-9	Majomfogó	40,-
076	C64	87/8-9	Többtényezös döntések	40,-
077	C64	87/8-9	Raszter	30,-
078	C16	87/10	Append	40,-
079	C64	87/10	Nagyitó	40,-
080	C64-H+	87/10	Makrók	40,-
081	C128	87/11	Struktúra	30,-
082	C128	87/11	Emlékeztetö	30,-
083	C64	87/11	Monitorka	40,-
084	C16	87/11	Cimröl-cimre	30,-
085	C64	87/12	Gyors scroll	40,-
086	C64	87/12	1541	40,-
087	C64	87/11-12	1541 + Monitorka	60,-
088	C16	87/12	Lokális változók	40,-
089	C16	87/12	Tekergö	40,-
090	C64	88/1	Fekete - fehér	20,-
091	C16	88/1	Datásitó	30,-
092	C64	88/1	Programkönyvtár	40,-
093	C16	88/1	Katasztrófa	30,-
094	VC20	88/2-3	TG BASIC	60,-
095	C64	88/2-3	Topy Typer	40,-
096	C64	88/2-3	Sprite editor super	60,-
097	C16	88/2-3	Katasztrófagép	30,-
098	C64	88/4	Modulgrafika	60,-
099	C64	88/4	Segitség!	30,-
100	C64	88/5	Guritsd a golyót!	60,-
101	PLUS/4	88/5	Egyenesrajzoló	30,-
102	PLUS/4	88/6	Szótár	60,-
103	C16	88/7-8	Gitáriskola	60,-
104	C64	88/7-8	Gyors BASIC	30,-
105	C128	88/7-8	Super hardcopy	40,-
106	C128	88/7-8	Grafikus nyomtatás	40,-
107	C64	88/9	Fast validate	60,-
108	C64	88/9	Adóelöleg	50,-
109	C64	88/9	Data bevivö	30,-
110	C64	88/9	DEF FN/X/	30,-
111	PLUS/4	88/10	Magyarositó	50,-
112	C64	88/10	Diszkkezelö	40,-
113	C16	88/10	DISASSEMBLER	30,-
114	C64	88/11	Négy képernyö	20,-
115	C64	88/11	Ablakok	20,-
116	C64	88/11	ASCII képernyö	20,-
117	C64	88/11	Pillanatfelvétel	20,-
118	C64	88/11	Blinky	20,-
119	C64	88/11	Szines villogtató	20,-
120	C64	88/11	80 karakteres üzem	20,-
121	C64	88/11	Villogtató	20,-
122	C64	88/11	Joystick lekérdezö	20,-
123	C64	88/11	Gyors sprite mozgatás	20,-
124	C64	88/11	DEF FN - másképpen	20,-
125	C64	88/12	Szuper input	40,-
126	C64	88/12	Ugráló szinsávok	30,-
127	C64	89/1	PRINT AT	20,-
128	C16	89/1	Funkcióbillentyük	20,-
129	C64	88/12	Minihelp	20,-
130	C64	89/2	Házi pók	40,-
131	C64	89/2	Csillagos ég	30,-
132	C64	89/1	Szalagról lemezre	40,-
133	C64	89/1	Kniffel	50,-
134	PLUS/4	89/1	Zombi	60,-
135	C64	89/1	Kétoldalas listázó	60,-
136	PLUS/4	89/1	Objektiv osztályozás	50,-
137	PLUS/4	89/2	Tömöritö	20,-
138	PLUS/4	89/2	Kifejtö	20,-
139	PLUS/4	89/3	3 - plusz - 1 magyarul	40,-
140	PLUS/4	89/3	Napló-kezelö	30,-
141	PLUS/4	89/3	Analóg óra	20,-
142	C128	89/3	Javitott renumber	40,-
143	AMIGA	89/4	Magyar karakterek	
144	C64	89/4	Keret csikozás	20,-
145	C64	89/4	Mozgó felirat	20,-
146	C64	89/4	26. sor	40,-
147	C16	89/4	Optinput	30,-
148	C16	89/4	Turbo-block	30,-
149	C16	89/4	Finom scroll	30,-
150	PLUS/4	89/4	Animáció	60,-
151	C128	89/4	Pót-funkciók	20,-
152	PLUS/4	89/5	Betüradar	40,-
153	C64	89/5	BHP virus keresö	40,-
154	C64	89/5	Képmutató	60,-
155	C128	89/5	VDC mint RAM floppy	30,-
156	C128	89/5	Joy SPRITE	30,-
157	C64	89/5	Javitott list	30,-
158	C64	89/5	Zenei segédlet	60,-

Alulírott megrendelem a következő programokat a Pötyögő Szolgálattól:

PROGRAM SORSZÁMA		ÁRA
1.	☐☐☐	☐☐ ,– Ft
2.	☐☐☐	☐☐ ,– Ft
3.	☐☐☐	☐☐ ,– Ft
4.	☐☐☐	☐☐ ,– Ft
5.	☐☐☐	☐☐ ,– Ft
6.	☐☐☐	☐☐ ,– Ft
7.	☐☐☐	☐☐ ,– Ft
8.	☐☐☐	☐☐ ,– Ft
9.	☐☐☐	☐☐ ,– Ft
Összesen:	☐☐ db	☐☐☐ ,– Ft

A programokat saját kazettámra kérem. ☐

A programokat saját lemezemre kérem. ☐
(a megfelelő kockát x-elje be!)

A programokat a Szolgálat által biztosított Parrot SS, SD lemezre kérem (190,– Ft/db) ☐☐☐

15' kazettára kérem. (30,– Ft/db) ☐☐

POSTA költség (25,– Ft) ☐☐ ,– Ft

Összesen: ☐☐☐ ,– Ft

A fenti összeget befizettem az Országos Commodore Egyesület számlájára:
OPT XIII., Visegrádi u. 7/b.
MNB 217-98292
OTP 565-3610

NÉV: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Cím: ....

EGYESÜLETI TAGSÁGI SZÁM: ☐☐☐

DÁTUM: ....

aláírás

Kérem, hogy a Commodore Újság legközelebbi számában jelenjen meg a következő szövegű apróhirdetés:
(minden oszlopba egy betűt írjon!)

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Név: ....

Egyesületi tagsági szám: ....

Feladandó az újság címére:
Commodore Újság Kárpát u. 7/a. I. em. 11.
1133

EGYESÜLETI TAGOK FÓRUMA

C 64

C 64 programokat cserélek lemezen. Keresem a DEFENDER OF THE CROWN című játékot. Listát kérek és küldök.
Matalik Krisztián, Gyöngyöspata, Szücsi út 11. 3035

Keresem a MERCENARY programot kazettán és más programokat is cserélek.
Székely Krisztián, Győr, ifjúság kört. 41. I/4.

Régi, nem szögletes dobozos, C 64-est, javításra szorulót vagy javíthatatlant keresek megvételre, ajánlatokat levélben, ár és hiba megjelöléssel kérem.
Kerekes Géza, Törökbálint, Tulipán u. 6. 2045

Színvonalas programokat cserélek kazettán. Listát minden esetben kérek.
Máté Miklós, Budapest, Csokonai u. 2. 1193

Keresem a GAME MAKER játékszerkesztő programot. Kb. 250 programom van. Listát kérek és küldök.
Fábián László, Dombóvár, VI. u. 28. 7200

Reset kapcsolót vennék olcsón!
Játékprogramokat cserélek, inkább kazettán, lehet lemez is.
Juhász Attila, Szombathely, Vak Bottyán utca 11.

Programcsere! Cserealap 2000 program. 5 ¼" lemezekért megegyezés szerint adok programokat.
Szabó Csaba, Ócsa, Üllői út 28. 2364

Keresem a MAUS 4 című programot, megvételre vagy cserére. Lehetőleg leírással.
Balló Attila, Budapest, Erzsébet Királyné útja 1/B. 1146

Programcsere! Csak színvonalas, 1988/89-es programok érdekelnek. Cserealapom: 1000 darab program. Keresem GIGA PAINT és GEO PUBLISH című programokat.

Programcsere kazettán! Telefon: (66) 41-246, 17 órától vagy hétvégén. Levélben.
Brád György, Tarhos, Béűési út 7/A. 5641

C 16, Plus/4

C 16-hoz 60K-s memóriabővítőt vagy kapcsolási rajzát keresem, cserére vagy megvételre.
Kaczmarek Edvárd. Telefon: 335-738

PLUS/4 programokat cserélnék kazettán. Keresem a SCOOPY DOO programot.
Pelyhe Szilárd, Budapest, Ördögárok utca 43. 1029

C 16, PLUS/4 programokat cserélek kazettán. Kb. 800 programom van. Listát kérek!
Sipos József, Kaposvár, Mező I. út 14. 7400

C 16, PLUS/4 programok cseréje, lemezen és kazettán! Listát kérek és küldök.
Raffai András, Budapest, Ady E. út 13—15. XI/114. 1211

C 16, PLUS/4 programokat cserélek kazettán. Lehetőleg új programok érdekelnek.
Boros Zoltán, Pánd, Arany J. utca 91/B.

PLUS/4 programcsere! Új sakkmásoló programot keresek. Jutányos áron eladó programok!
Karmazin Zoltán, Szolnok, Pf. 137, 5001

PLUS/4 programokat cserélek kazettán vagy lemezen. 700 programom van. BASIC programok nem érdekelnek! Listát kérek!
Hollik Zsolt, Debrecen, Vukovics Sebő u. 1. 4030

Vegyes

AMIGA programok és információk cseréje.
Bárdos Ferenc, Kalocsa, Miskei út 20. 6300

Commodore PC-vel kapcsolatos minden információ érdekel.
Kovács Mihályné, Budapest, Néphadsereg u. 14. 1055

Vadonatúj ENTERPRISE 128 nagyon olcsón eladó.
Balázs Jánosné. Telefon: 573-135

VC 20-as géppel rendelkezők ismeretségét keresem.
Pázsy István, Budapest, Dembinszky utca 46. IV/61.

1541-es vagy 1541-II-es diszk drive-ot veszek. Lehet használt is.
Lajtos Máté, 4032 Debrecen, Martonfalvi u. 14. fsz. 2.

Eladó kétkazettás AKAI HX—A301 magnó deck.
Karmazin Zoltán, Szolnok Pf. 137. 5001

ATARI 1040 alapgépet, floppyt keresek!
Bali József. Telefon: 420-212.

kedvezmények

A NOVOTRADE RT. 2C Áruházában az Egyesület PLUSZ- és SZUPER PÁHOLYÁNAK tagjai kedvezménnyel vásárolhatják meg a következő programokat:

A kedvezmény a megjelenéstől számított egy hónapig érvényes.



DEÁKPÁHOLY: –20%

Basic, Logo, Pascal	149,— Ft	119,20 Ft
Fizikomp	129,— Ft	103,20 Ft
Első könyvem CHIP	99,— Ft	79,20 Ft
Sakkprogramozás mindenkinek	239,— Ft	191,20 Ft
Robotok és a C—64	249,— Ft	199,20 Ft

PLUSZPÁHOLY: –10%

Mesterséges intelligencia	349,— Ft	279,20 Ft
Prolóóban programozni könnyű	219,— Ft	175,20 Ft
MPS 1000 Felhasználói kézikönyv	180,— Ft	144,— Ft

SZUPERPÁHOLY: –20%

PRINTER BASIC	8459,— Ft	6767,20 Ft
Mesterséges megvilágítás	4750,— Ft	3800,— Ft

KEDVEZMÉNYEK a PC-Szalontól a Szuperpáholy tagok részére!
IMPOSSIBLE MISSION
II. 15% 781,– ÁFÁVAL
játékprogram
AMBASSADOR PC szövegszerkesztő
10% 5950,– ÁFÁVAL

Ápisz

ÉRVÉNYTELEN

júniusi
60 forintos
vásárlási utalvány
Beváltható készpénzes vásárlás esetén az ÁPISZ szaküzleteiben
XI., Budafoki út 7.
VIII., Szigony u. 15.
Érvényes: 1989. szeptember 30-ig

NOVOTRADE

ÉRVÉNYTELEN

júniusi
60 forintos
vásárlási utalvány
Beváltható készpénzes vásárlás esetén a 2C áruházban XIII., Balzac u. 35. és a Művelt Nép vidéki boltjaiban működő 2C sarkokban.
Érvényes: 1989. július 31-ig

10%

A Newline számítástechnikai vállalkozás 10% kedvezményt ad az egyesület tagjainak:

C 16 beépíthető 64 KByte memóriabővítő	1990,– Ft
16–64-es átkapcsoló	150,– Ft
beépítés munkadíja	490,– Ft
ROMTURBO 16	770,– Ft
együttes megrendelés esetén	3400,– Ft
árengedménnyel:	3060,– Ft

Jogosultak: a *Plusz- és a Szuperpáholy tagjai*
Igazolás: *ennek a tikettnek postai elküldésével*
Cím: Newline, 1014 Budapest, Tárnok u. 26. 1/5.

NEWLINE
HARDWARE SOFTWARE

10%

A Fotoelektronik-Novotrade-Alfa KFT az alább felsorolt szervizeiben mindenféle szervizszolgáltatás munkadíjából és az 1989. évben megkötött átalánydíjas szerződés összegéből 10% kedvezményt ad az egyesületi tagoknak.
Határidő: nincs, illetve átalánydíjas szerződésnél 1989. 12. 01.

A kedvezményt nyújtó szervizeink:

1053 Budapest, Magyar u. 12—14.	Telefon: 173-551
1083 Budapest, Szigony u. 9.	Telefon: 343-153
1191 Budapest, Gábor Á. sétány 3.	Telefon: 274-763
3100 Salgótarján, Arany J. u. 3.	Telefon: 32-14-007
3525 Miskolc, Fazekas u. 1—3.	Telefon: 46-17-011
4034 Debrecen, Holló L. u. 14.	Telefon: 52-32-863
5600 Békéscsaba, Bartók B. u. 37.	Telefon: 66-27-195
6726 Szeged, Csongrádi sugárút 76.	Telefon: 62-13-377
7400 Kaposvár, Füredi u. 24.	Telefon: 82-16-307
7624 Pécs, Jurisics M. u. 17.	Telefon: 72-11-812
8000 Székesfehérvár, Széchenyi u. 15/a.	Telefon: 22-12-711
9024 Győr, Bem J. tér 1.	Telefon: 96-12-802
9700 Szombathely, Szalonok u. 31.	Telefon: 94-13-419

FOTOELEKTRONIK NOVOTRADE

Igazolás: a javítandó berendezés leadásakor egyesületi igazolvánnyal.
A kedvezmény többször is igénybe vehető.

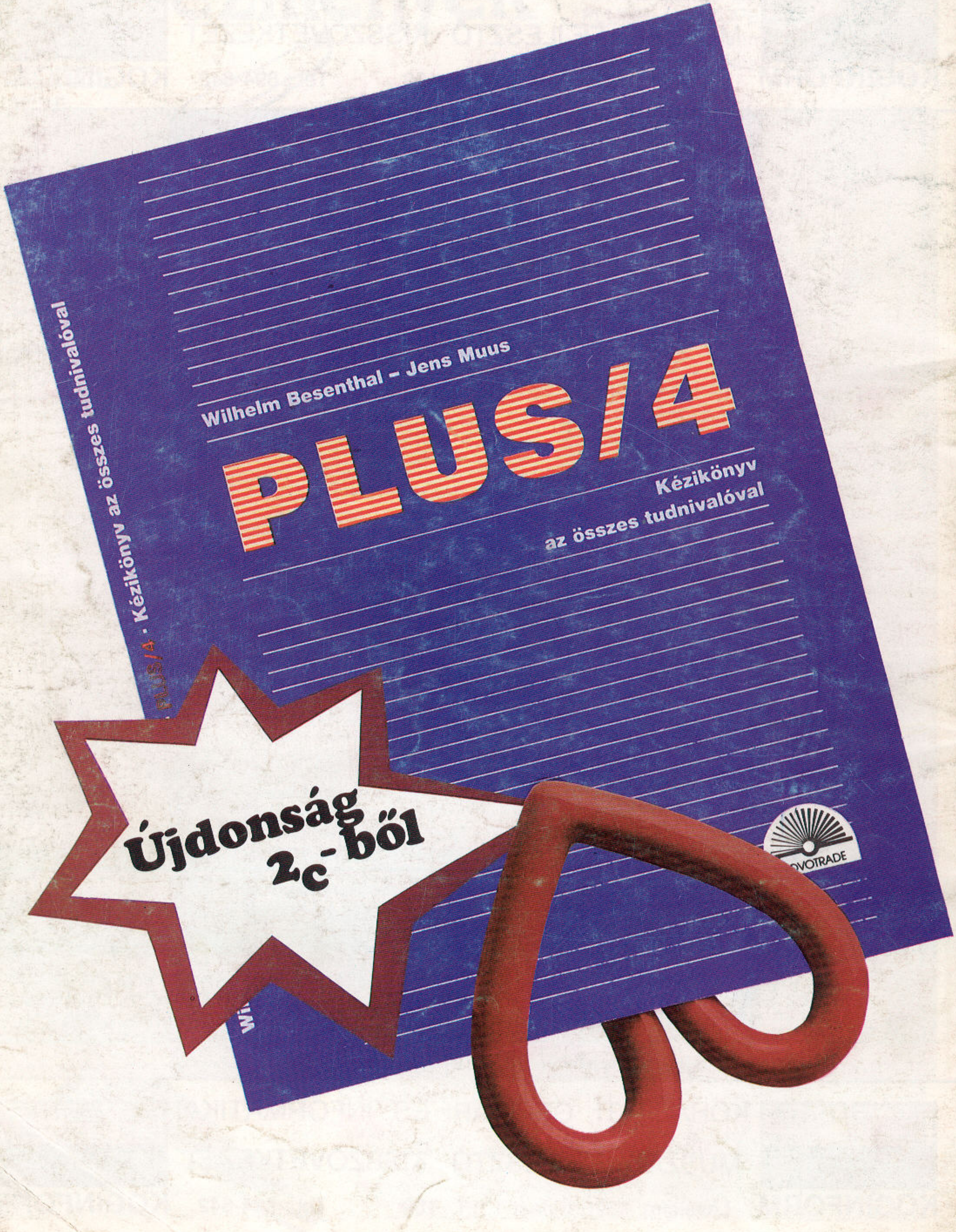
KOGINFORM

KOHÁSZATI, GÉPIPARI ÉS INFORMATIKAI
MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET

Levélcím: 1325 Újpest 1. Pf. 159. Tel.: 894-642

KOGINFORM

Ára: 48 Ft


Wilhelm Besenthal – Jens Muus
PLUS/4
Kézikönyv
az összes tudnivalóval
PLUS/4 · Kézikönyv az összes tudnivalóval
Újdonság
2c-ből